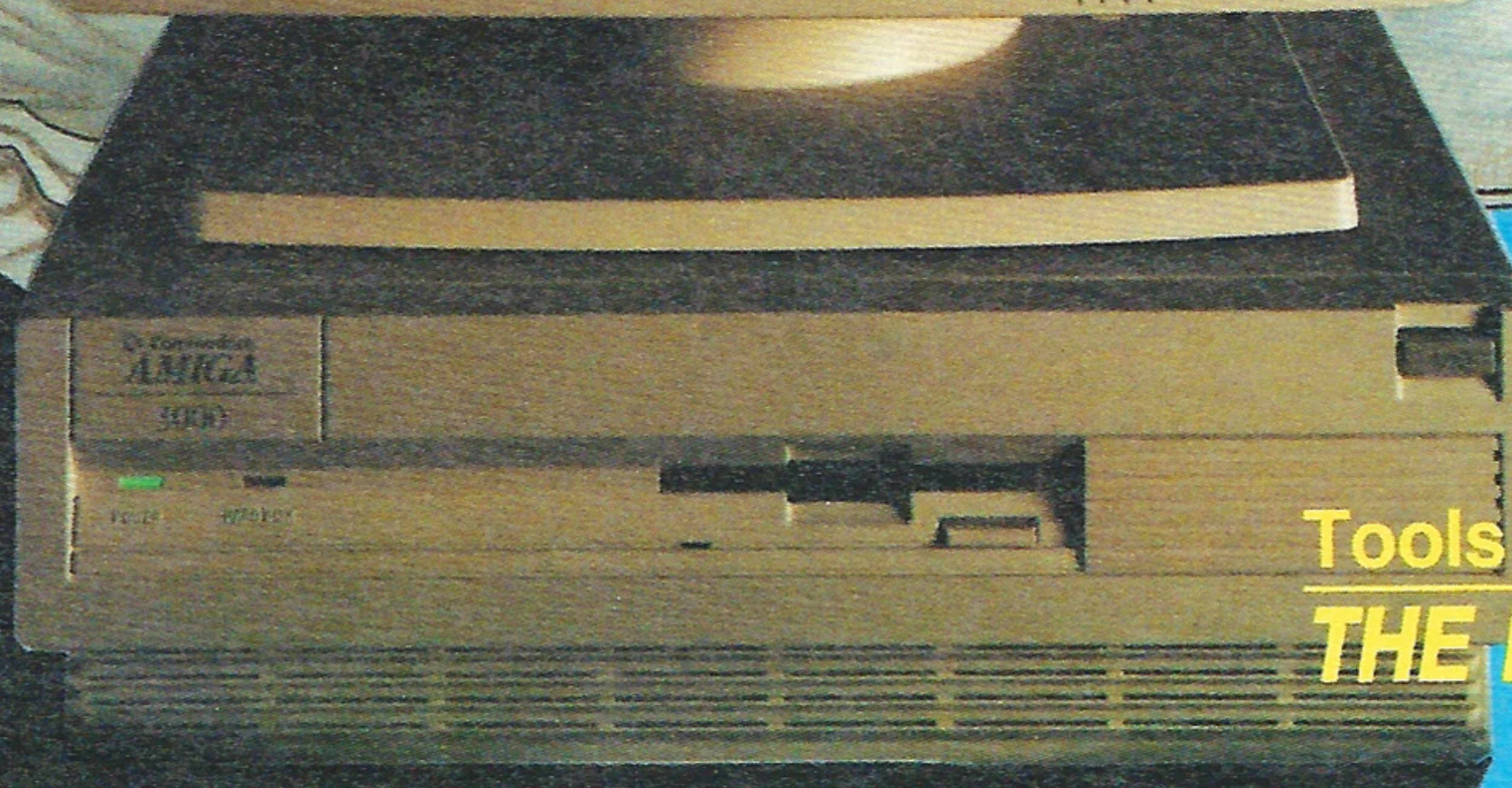
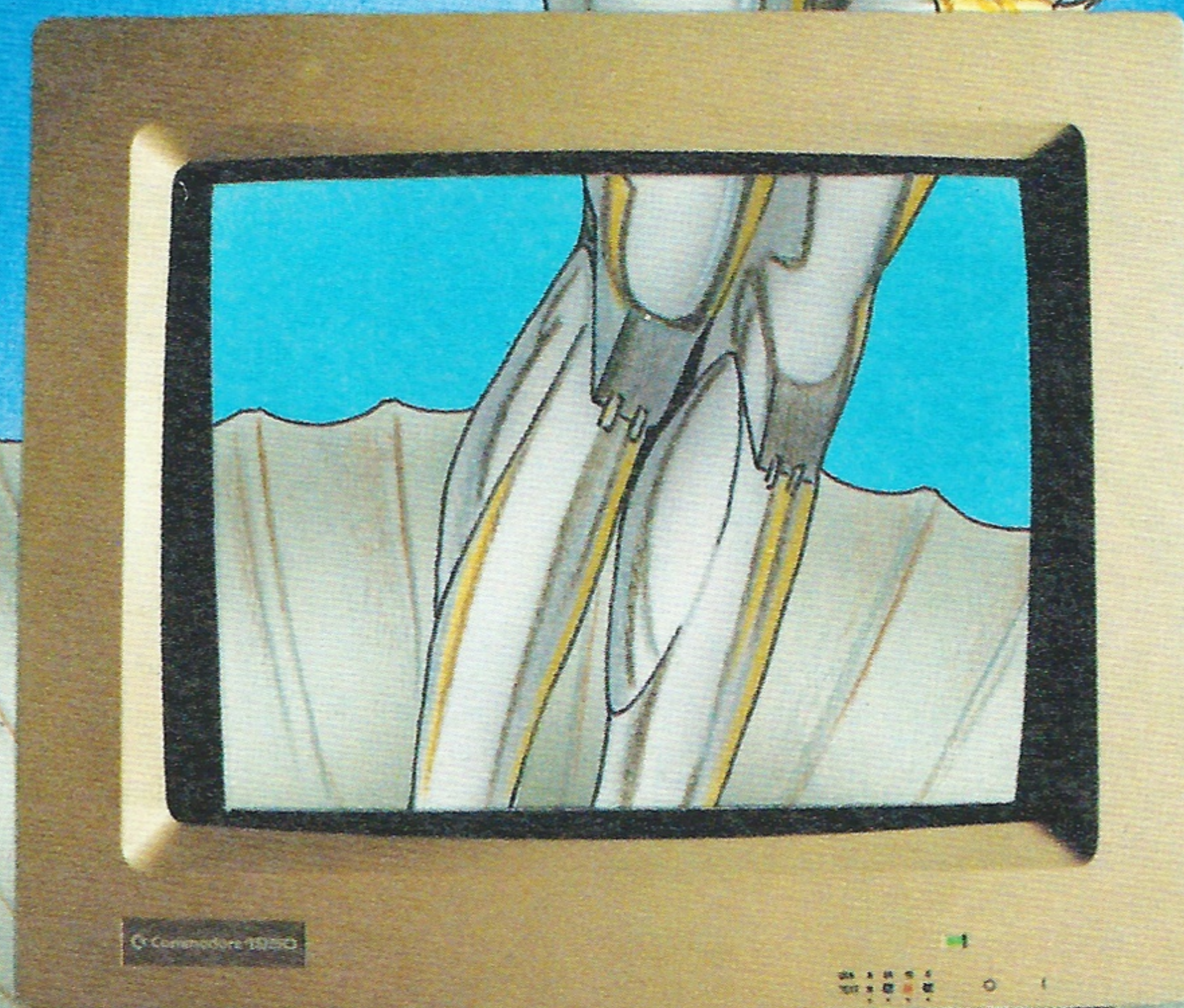
N. 33/1991
Sped. in abb. post. gr. III/70
L. 15.000
AMIGA
BYTE
68000
by Elettronica 2000
SUL DISCO
ALERT I GURU PERSONALIZZATI
ZOOM NUOVO COMPATTATORE DI DISCHI
AUTOMALAB SIMULAZIONE CELLULARE
AMIGATRATION HAI BUONA MEMORIA?
TRIPPPIN INEDITO GIOCO DI STRATEGIA
MOVESYS ASSEGNA IL VOLUME SYS:
EL MASO STUDIO DI FUNZIONI
RRAMDISK IL BOOT DAL DISCO RAM
IMPLODER POTENTE TURBO CRUNCHER
MISSILE CMD ATTACCO DAL CIELO
Software
IL MEGLIO DEL PUBBLICO DOMINIO
Grafica e Matematica
MATHVISION & VISTA
Utility
PERSONAL FONTS MAKER
TIPS & TRICKS
Linguaggi
COMPILATORE, EDITOR E LINKER DEL BENCHMARK MODULA 2
Assembler
CICLI E SALTI CONDIZIONATI
Hardware
SUPRA 500XP HARD DISK
Tools
3D TEXT ANIMATOR 1.2
PROFESSIONAL PAGE 2.0
Tools grafici
THE DIRECTOR 2.0

Direttore
SIRA ROCCHI

Direzione Editoriale
MARIO MAGRONE

Direzione Tecnica
GIANCARLO CAIRELLA

Segreteria di Redazione
SILVIA MAIER

Grafica
NADIA MARINI

Fotografie
MARIUS LOOK

Disco a cura di
VITTORIO FERRAGUTI

Copertina
GIGI MURATORE

Collaborano ad AmigaByte: Francesco Annoni, Luca Arienti, Laura Baricevic, Paolo Bozzo, Luca Brigatti, Marco Brovelli, Paolo Colombo, Antonio De Lorenzo, Enrico Donna, Enrico Frascati, Renato Grossi, Fabrizio Lodi, Silvia Malaguti, Vincenzo Marangoni, Dario Martinelli, Luca Mirabelli, Pierluigi Montanari, Lorenzo Orlandini, Domenico Pavone, Graziano Pavone, Roberto Pellagatti, Riccardo Premoli, Guido Quaroni, Fabio Rossetti, Giuseppe Sacchi, Emanuele Scribanti, Paolo Sisti, Leonardo Tennozio, Aurora Tragara, Vertigo.

Redazione
C.so Vitt. Emanuele 15
20122 Milano
tel. 02/795047
ore pomeridiane.
Per telefonate tecniche: solo mercoledì h 15-18.

Amministrazione, Redazione, Pubblicità: L'Agorà srl: C.so Vittorio Emanuele 15, 20122 Milano. Fotocomposizione: Compostudio Est, selezioni colore e fotolito: Eurofotolit. Stampa: Garzanti Editore S.p.A. Cernusco S/N (MI). Distribuzione: SO.DI.P. Angelo Patuzzi spa, Via Zuretti 25, Milano. Amiga Byte è un periodico mensile registrato presso il Tribunale di Milano al n. 215 il 29 marzo 1988. Resp. Sira Rocchi. Spedizione in abbonamento postale Gr. III/70. Pubblicità inferiore al 70%.

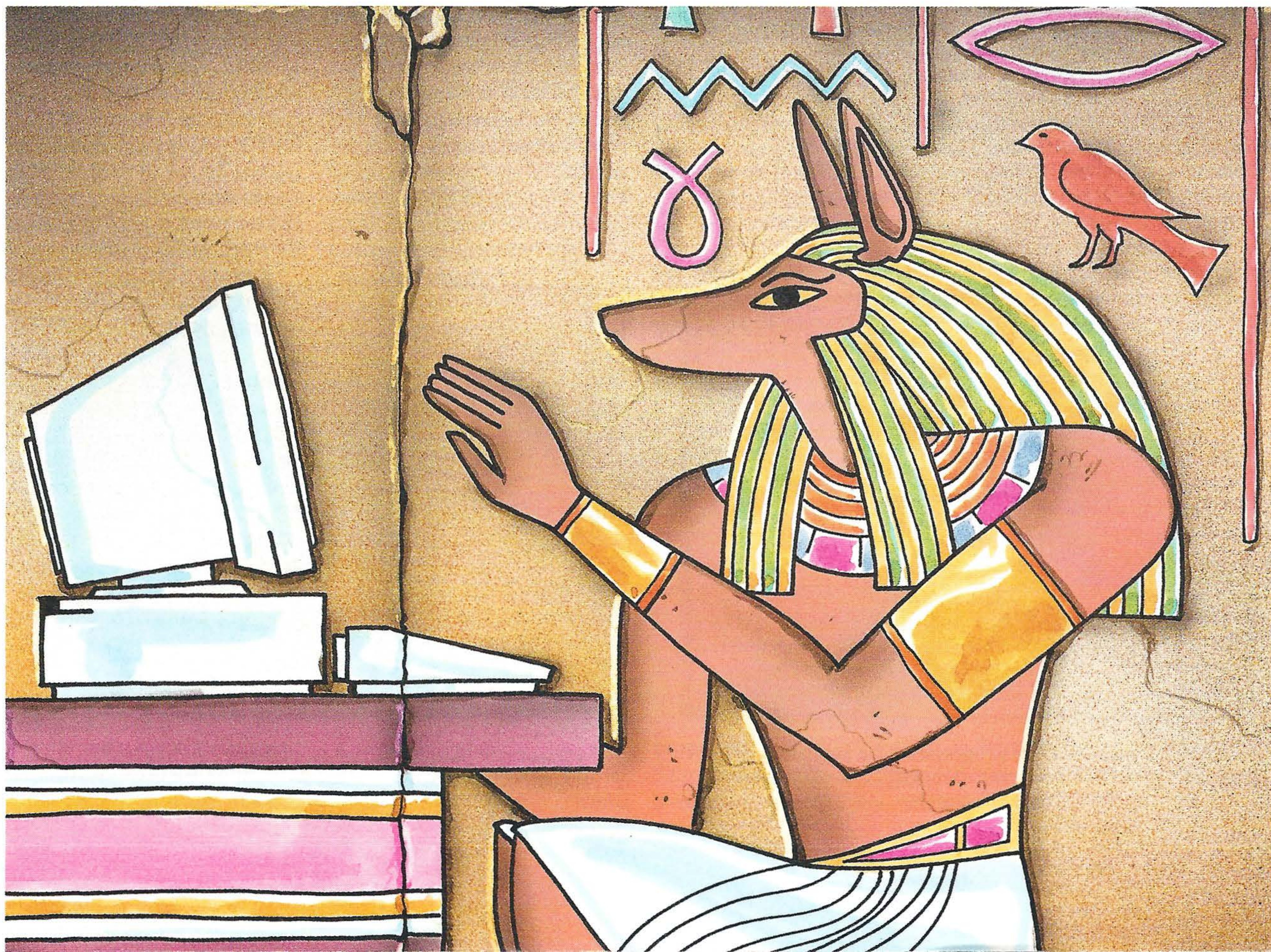
IL MENU

Personal Fonts Maker

Dalla software house del popolare «C1-Text», un programma di disegno di font per sfruttare al massimo le potenzialità della vostra stampante.

di ANTONIO DE LORENZO

Amiga, come tutti i computer, visualizza sullo schermo i caratteri del testo sotto forma di una matrice di punti (**bitmap**). Il sistema operativo è predisposto per leggere il set di caratteri da utilizzare nella **directory Fonts:** o, nel caso in cui non sia possibile, direttamente dalla Rom. Un set di caratteri (**font**) Amiga è composto da 256 caratteri di cui uno, detto **indefinito**, di riserva, che il sistema operativo ed i programmi utilizzano quando uno o più caratteri del set non sono definiti.

L'uso di caratteri diversi è un aspetto importantissimo nella comunicazione scritta: può migliorare la leggibilità di un testo, aiutare a tener desta l'attenzione del lettore, conferire un aspetto personale a quanto redatto. È per questo che una delle necessità più sentite dagli utenti è la disponibilità di font nuovi visualizzati su video e, soprattutto, di caratteri diversi in fase di stampa. Per stampare con font diversi

COS'È UN FONT?

In gergo tipografico ed informatico si definisce font l'insieme dei caratteri alfanumerici, dei segni di interpunzione e degli altri simboli semigrafici normalmente utilizzati per la stampa o la visualizzazione a video dei testi. Un font possiede determinate caratteristiche: il primo attributo è l'altezza in pixel (definita, in ambito tipografico, corpo) indicata da un numero che, in genere, segue la denominazione del font (ad esempio Ruby 12). La larghezza varia tra minuscolo e maiuscolo e inoltre ci permette di introdurre un'ulteriore distinzione tra caratteri a larghezza fissa (tutti con la medesima larghezza) e tra caratteri proporzionali come la maggior parte dei font Amiga dove ogni carattere occupa lo spazio necessario ad una sua chiara visualizzazione (per esempio la lettera «i» occupa meno spazio di una lettera «m»). I font posso-

!"#$%&'()*+,-./0123456789:;<=>?
@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`abcdefghijklmnopqrstuvwxyz{|}~

!"#$%&'()*+,-./0123456789:;<=>?
@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`abcdefghijklmnopqrstuvwxyz{|}~

!"#$%&'()*+,-./0123456789:;<=>?
@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`abcdefghijklmnopqrstuvwxyz{|}~

Prova di stampa, eseguita con una stampante a 24 aghi, di alcuni font acclusi a «Personal Font Maker». Le lettere che compongo il terzo font (Perseus) sono state acquisite tramite uno scanner da un manoscritto del XVI secolo, ritoccate con un programma pittorico, poi importate in «PFM».

no possedere o meno delle grazie (tratti terminali delle lettere, a volte detti serif), possono essere in forma di apici e pedici (cioè in dimensioni ridotte stampati in alto o in basso rispetto alla posizione dei caratteri presenti in una riga). I caratteri possono essere rappresentati con stili diversi come il corsivo, il neretto o il sottolineato, eventualmente combinati tra loro. Tutte le varianti di un font vengono raggruppate in una famiglia che in genere riporta il nome tipografico (es. Times, Helvetica, Topaz, Garamond, Granion). Il font, residente nella Rom, normalmente visualizzato da Amiga è il Topaz 8, non proporzionale.

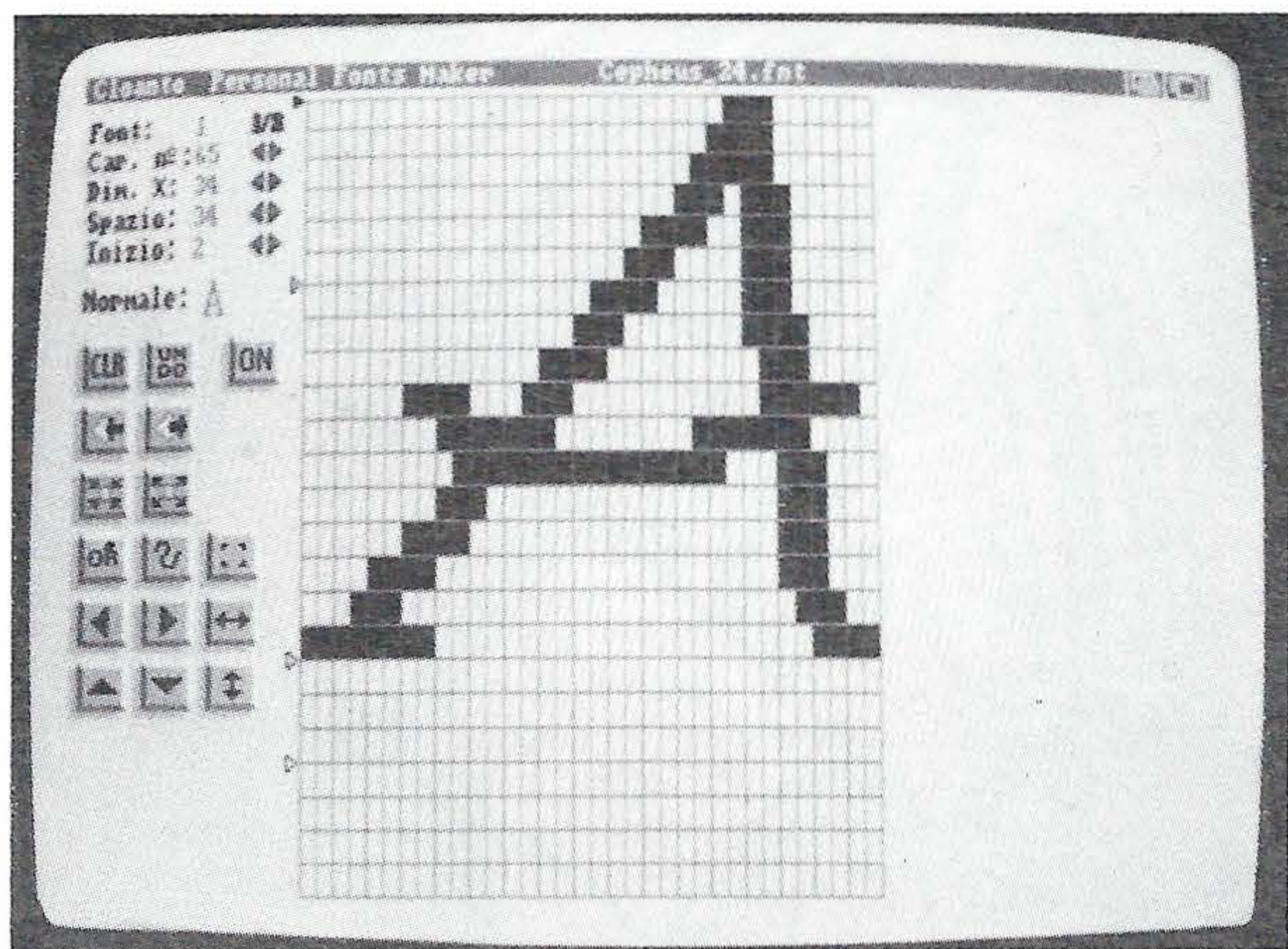



Fig. 1: l'ambiente del programma. La maggior parte dello spazio è occupata dalla griglia della zona di editazione.

da quelli disponibili da stampante, fino ad ora l'utente era obbligato a comprare spesso non economiche **Font Card** da inserire (nelle stampanti che lo consentono) in uno slot predisposto.

DISEGNARE UN FONT

Il programma «**Personal Fonts Maker**» (distribuito da Cloanto, Via G.B. Bison 24, 33100 Udine al prezzo di 99.000 Lire) consente, oltre che la progettazione, il disegno e la manipolazione dei set di caratteri da utilizzare sul video, di trarre vantaggio da una caratteristica delle stampanti spesso non adeguatamente sfruttata. La maggior parte di esse dispone infatti di una quantità di memoria Ram in cui l'utente può trasferire i propri font e riprodurli in fase di stampa alla stessa velocità e qualità dei font residenti. La gestione di questa peculiarità non è però delle più semplici (sia per le caratteristiche intrinseche delle stampanti, sia per la diversità di gestione mai unificata in uno standard, etc.). Grazie a «PFM» (così chiameremo d'ora in avanti il programma per brevità) questa operazione viene semplificata e risulta alla portata di tutti.

«PFM» è composto da due dischi (contenenti il programma principale, una serie di esempi, macro ed utility assortite), e da un manuale di 330 pagine, interamente in italiano e particolarmente chiaro e completo. Il programma funziona su qualsiasi modello di Amiga, compresi i 3000 dotati del nuovo sistema operativo 2.0, e può operare anche con soli 512 Kb, anche se è consigliabile disporre di almeno 1 Megabyte poiché, se scarseggia la Ram, «PFM» disattiva l'audio, chiude il Work Bench, riduce il numero di colori dello schermo e prende altri provvedimenti per raggranellare quanta più memoria possibile.

Per i possessori di un hard disk è presente un file denominato «**InstallPFM**» che provvede all'installazione automatica di «PFM» e di tutti i file necessari al suo corretto funzionamento.

Una volta lanciato, il programma presenta, al centro dello schermo, una fitta griglia denominata **zona di editazione** (Fig. 1), mentre il puntatore del mouse si trasforma da freccia in croce quando percorre tale zona. Utilizzando il mouse si possono selezionare i punti della griglia semplicemente tramite la pressione del tasto sinistro, mentre con il destro il punto sottostante il puntatore viene cancellato.

«PFM» può (memoria permettendo) operare su due font contemporaneamente, denominati **Font1** e **Font 2.** Sul lato sinistro dello schermo sono presenti alcuni gadget modificabili attraverso il mouse, che il manuale definisce **bottoni.** Al **parametro Font** (primo della lista) sono associati due bottoni: il primo indica la selezione del font 1 o 2, a seconda della cifra che vi si digita manualmente, o tramite la selezione del bottone con la dicitura **1/2.**

Il **testo Car. n** è seguito da tre bottoni; anche in questo caso un carattere può essere selezionato semplicemente inserendone la cifra dopo aver cliccato nella zona con il mouse, o procedendo sequenzialmente in avanti o indietro con le due frecce contrapposte. **Dim. X** controlla la larghezza del carattere; se il font non è proporzionale, questo parametro è costante per ciascun carattere. **Spazio** indica il numero di punti di cui va spostata, verso destra, la posizione di stampa dopo che il carattere è stato stampato. Questo parametro influen-

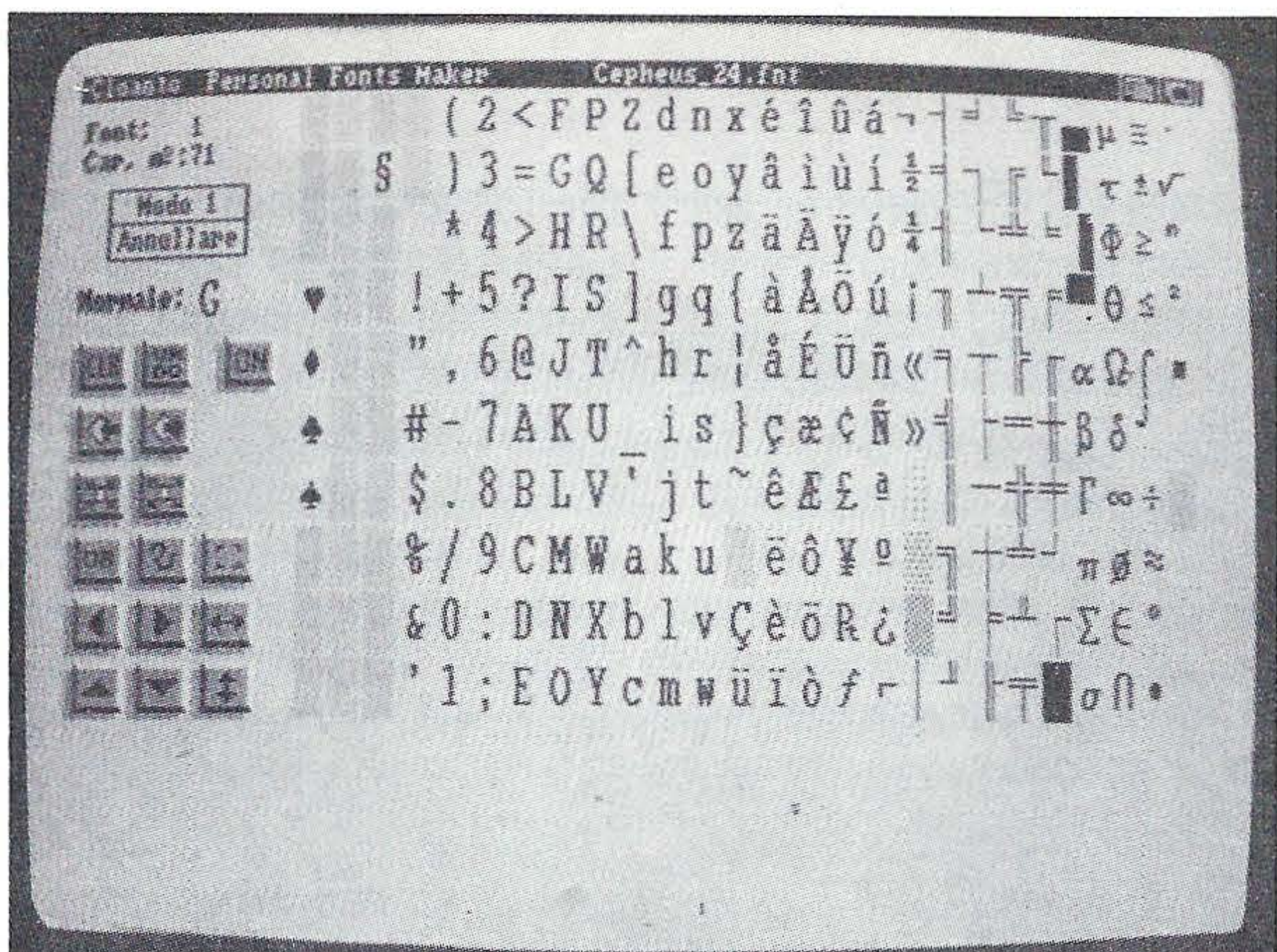



Fig. 2: un esempio di attivazione del modo «selezione veloce» di «PFM».

za la posizione del carattere successivo, e di solito il suo valore è appena superiore a quello di **Dim X.** Ad esempio, se Dimensione X è pari a 36 e Spazio è uguale a 40, questo significa che a destra di ogni lettera c'è uno spazio di 4 pixel.

Il **bottone Inizio** determina invece il numero di pixel che separano il carattere da quello alla sua sinistra, ed accetta valori compresi tra −128 e 127.

Infine, il **bottone Normale** non contiene l'immagine ridotta di quanto rappresentato nella zona di editazione, bensì quella standard associata a ciascun codice del set di caratteri. Quando questo bottone viene selezionato, si attiva il **modo selezione veloce** e viene mostrata una tabella di caratteri (Fig. 2). Con il mouse si può selezionare una delle immagini standard del font: questo costituisce il modo di operare detto **Modo 1.** In alto a sinistra si può selezionare il **Modo 2**, dove ad ogni carattere disattivato è sovrapposta la scritta **OFF**. Per attivarli è sufficiente clickarvi sopra con il tasto destro del mouse.

LA STANZA DEI BOTTONI

La colonna di sinistra è completata da altri 16 bottoni privi di denominazione ma con una serie di simboli grafici sul dorso che ne indicano la funzione. Il **bottone CLR** cancella quanto contenuto nella zona di editazione, mentre **UNDO** consente di tornare alla situazione precedente annullando l'ultima operazione. Sulla stessa fila è presente il **bottone Stato di Attivazione** che possiamo selezionare in **ON** e **OFF** per decidere quali caratteri escludere da un determinato font (il disegno nella zona di editazione viene interdetto con questo bottone in stato di OFF). La griglia della zona di editazione con l'eventuale carattere contenuto cambierà colore (verde chiaro/scuro) a seconda dello stato definito. Il programma non salva i dati dei caratteri disattivati, anche se le loro caratteristiche vengono conservate in memoria fino al termine della sessione di lavoro.

Nella seconda riga troviamo altri due bottoni con il disegno di due frecce, ognuna di verso opposto. Il primo bottone memorizza il carattere ed i relativi parametri in una zona di memoria predisposta (**buffer**). Quanto in essa contenuto può essere richiamato con il secondo **bottone «Richiamare carattere»**.

In terza fila sono presenti altri due bottoni di disegno (quattro freccine che puntano verso un punto e quattro che puntano verso l'esterno) e funzione contrapposta. Il primo è detto **bottone Ridurre**, il secondo **Ingrandire** ed entrambi agiscono, come si può intuire, nel diminuire o aumentare le dimensioni della zona di editazione. In quarta fila sono presenti tre bottoni. Il primo, detto **OR**, attiva questo stato ed agisce nelle funzioni di richiamo carattere, di editazione del pennello, di spostamento e di capovolgimento che vedremo in seguito. Attivato questo modo e selezionata una delle funzioni elencate, si esegue un'operazione logica di OR tra i nuovi dati del carattere e quelli visualizzati.

Di seguito troviamo il **bottone Penna/Pennello.** Per mezzo del mouse è possibile disegnare nella zona di editazione in due modi diversi: facendolo in **modo penna** si utilizza un unico punto (pixel), mentre in **modo pennello** ci si può avvalere di un brush definito in precedenza o caricato. Sempre con il mouse, in uno di questi modi si può disegnare tramite il tasto sinistro, o cancellare tramite la pressione del tasto destro. Il terzo bottone della riga (**Definire pennello**) riporta simbolo e funzione uguali all'operazione di taglio di un brush di «**DPaint**». Esso serve infatti per ritagliare una parte del contenuto della zona di editazione, utilizzando il mouse.

CAPOVOLGIMENTO E SPOSTAMENTO

Nelle ultime due file sono presenti quattro bottoni (**Spostamento**) con relativi cursori orientati. La selezione di un determinato bottone sposta il contenuto della zona di editazione nella direzione indicata dalla freccia. Infine, i due bottoni **Capovolgimento**, a destra di quelli **Spostamento**, definiti da linee orizzontali e verticali con due frecce contrapposte, capovolgono orizzontalmente o verticalmente i dati presenti nella zona di editazione.

Un valido ausilio per chi si appresta a disegnare caratteri è il posizionamento di quattro punti di riferimento che contrassegnano la zona di editazione. Per fissare questi punti bisogna spostarsi con il mouse a sinistra nella zona definita come **zona dei punti di riferimento**. Qui il puntatore cambia la sua rappresentazione in due frecce verticali contrapposte. Premendo il tasto sinistro del mouse si attiva il **modo punti di riferimento** e, in alto a sinistra dello schermo, appaiono i tasti funzione (da **F1** a **F4**) associati ad ogni punto. Premendo uno di questi tasti si spostano i punti di riferimento. Ogni punto di

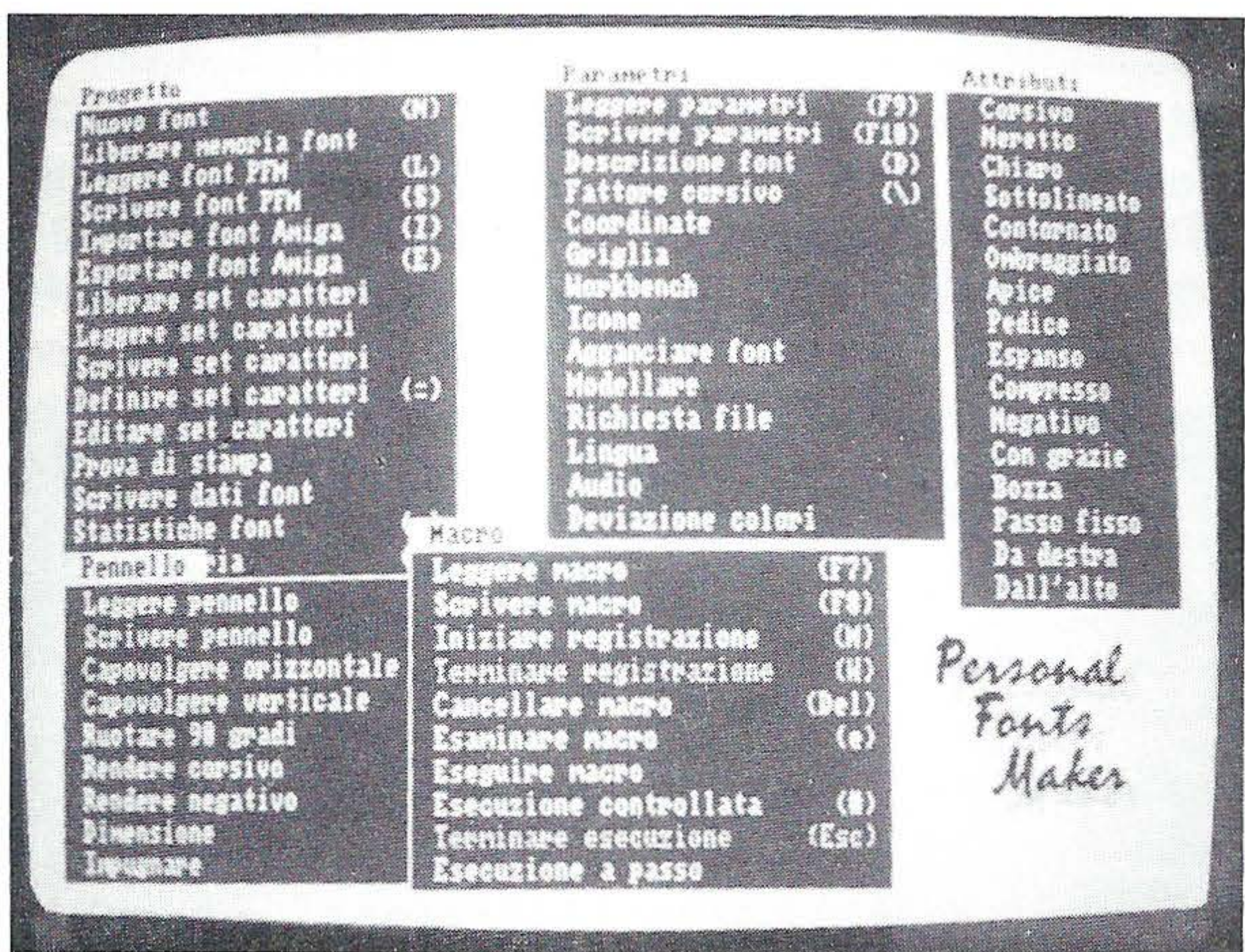



Fig. 3: i cinque menu del programma comprendono un gran numero di opzioni, fortunatamente tutte in italiano.

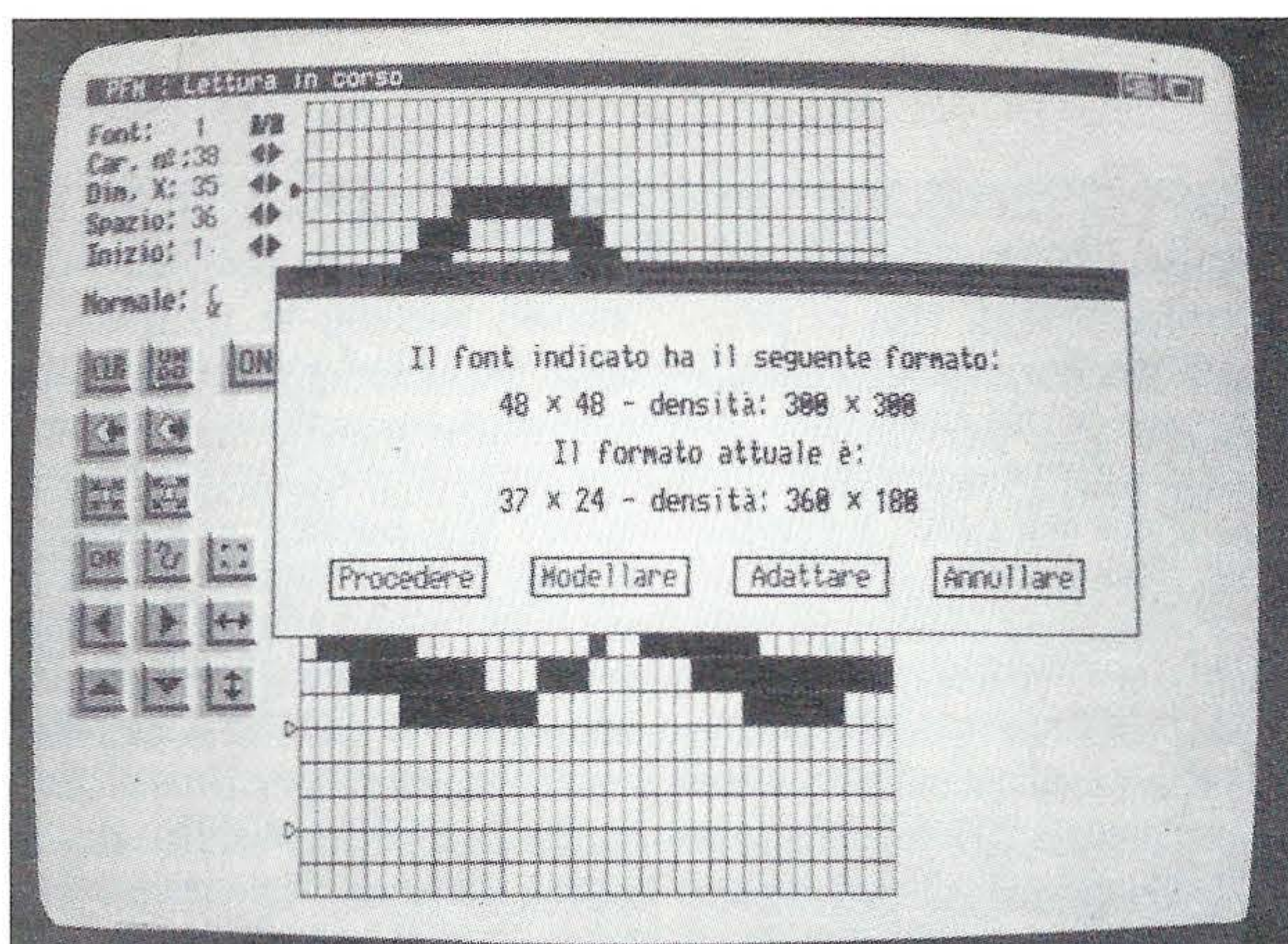



Fig. 4: il requester che appare quando, durante la lettura di un font in formato «PFM», il programma riscontra dimensioni diverse da quelle del font corrente.

riferimento viene segnalato mediante una freccia triangolare visualizzata nel lato sinistro della zona di editazione.

IL MENU PROGETTO

«PFM» comprende cinque diversi menu, in lingua italiana, pieni di opzioni (Fig. 3). Il primo di essi, denominato **Progetto**, elenca le operazioni di memorizzazione e di salvataggio dati, oltre che fornire all'utente tutta una serie di informazioni circa lo stato dei font e del programma. Prima di passare all'analisi di ogni singolo comando si rendono necessarie alcune premesse per ciò che attiene a tali operazioni.

«PFM» può sia leggere che scrivere font in due formati (Cloanto Personal Font Maker o CPFM, e Amiga bitmap), nonché generare font in formato specificato dall'utente o presente in uno dei file parametri (per esempio, Set caratteri IBM e compatibili). Con «PFM» inoltre si possono creare file dati in formato definito dall'utente tramite il linguaggio FFDL (dall'inglese *Font Format Description Language*, cioè «linguaggio di descrizione del formato font» e si possono stampare font con qualsiasi modello di stampante, o creare set di caratteri per programmi diversi da utilizzare in altri tipi di calcolatori.

La prima opzione che incontriamo è **Nuovo font**, la quale cancella il font corrente e ristabilisce i parametri standard. Il programma avverte con un Alert nel caso l'operazione cancelli i dati di un font non ancora memorizzato. **Liberare memoria font** libera dalla memoria un eventuale secondo font lasciandone operativo almeno uno. **Leggere font PFM** carica un font memorizzato in formato PFM. In caso di caricamento di un font in formato diverso da quello che si intende caricare, il programma avverte con un apposito quadro (Fig. 4) del formato attuale e di quello che si intende importare. Questo quadro presenta in basso quattro bottoni: **Procedere** carica il font senza modificare i dati grafici; **Modellare** modella il nuovo font in base a quello corrente; **Adattare** modifica il formato del font corrente adattandolo a quello del font selezionato. Infine, **Annullare** sospende l'operazione e **Scrivere font PFM** salva il font corrente tramite un apposito request.

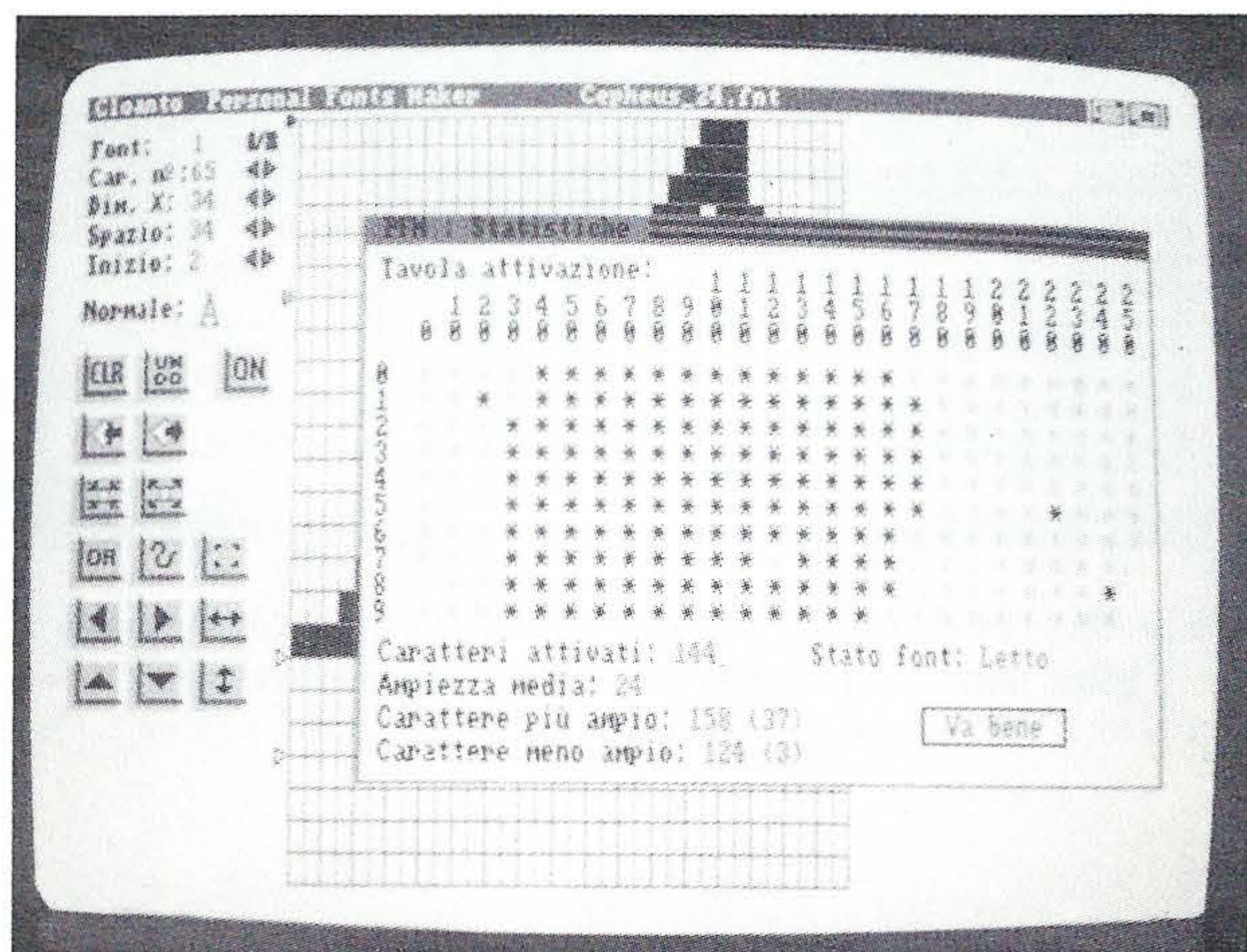
Importare/Esportare font Amiga leggono e salvano font in formato Amiga. **Liberare set caratteri** libera la memoria dal contenuto dei dati del font corrente. **Leggere/Scrivere set caratteri** permette di leggere nonché di salvare diversi set di caratteri tra cui set Amiga, PC, e molti altri. **Definire set caratteri** modifica i parametri del vettore codifica. Poiché, come abbiamo visto, una delle peculiarità di «PFM» è quella di poter utilizzare due font contemporaneamente, ognuno con la possibilità di appartenere ad un set diverso, tale vettore consiste in un insieme di immagini standard che permette a «PFM» di scambiare dati tra font Amiga e font basati su set di caratteri differenti. Tale opzione richiama un requester per l'introduzione sia dei caratteri che dell'apposito codice di conversione. **Editare set caratteri** è usato insieme all'opzione precedente per modificare o per creare un nuovo font, e consente la modifica delle immagini standard dei font presenti nel set.

L'opzione **Prova di stampa** contiene quattro sotto-opzioni per inviare alla stampante un singolo carattere quattro volte (e vedere così l'aspetto del carattere singolo ed inserito tra alcune lettere). Si può scegliere se stampare un solo carattere (**Carattere**) oppure l'intero font (**Font**), o la porzione di quest'ultimo specificata nel request del **menu Parametri/Descrizione font.** I restanti due comandi eseguono le operazioni omonime (**Avanzamento riga** e **Avazamento pagina**) sulla stampante.

Scrivere dati font consente di inviare i dati di un font o in un **File** oppure direttamente alla **Stampante.** Queste due sotto-opzioni sono particolarmente importanti dal momento che contengono i dati dei font che la stampante deve memorizzare. In particolare, il salvataggio del font in un file serve ad impiegare quel font insieme ad altri programmi di scrittura (word processor o editor) senza eseguire di nuovo «PFM» ma utilizzando un'utility presente nei dischetti del programma. **Statistiche font** visualizza (Fig. 5) una serie di notizie sul font corrente. **Stato memoria** elenca la porzione di Chip e di Fast memory occupata e libera. **Cancellare** è un comando utile per eliminare un qualsiasi **File** o direttamente un **Font** Amiga. **Informazioni**, infine, visualizza un quadro con notizie circa la versione in uso ed altre note riguardanti il programma, mentre **Fine Operazioni** termina l'esecuzione del programma non prima comunque di aver avvertito l'utente di una eventuale presenza di dati non salvati.

IL MENU PENNELLO

Le opzioni contenute in questo menu riguardano la

Fig. 5: il riquadro delle statistiche mostra informazioni sulle dimensioni massime, minime e medie dei caratteri del font.

LE UTILITY DI PERSONAL FONT MAKER

Completano il pacchetto «PFM» una serie di interessanti utility dedicate alla gestione dei driver per stampante e dei font.
È utile sollecitare il ricorso al manuale della stampante per la verifica e la modifica di eventuali parametri discordanti, e la selezione dell'appropriato driver di stampa da WorkBench (sezione Preferences) per assicurare un corretto trasferimento di dati dal calcolare alla stampante. Sovente però insorgono problemi anche dopo la selezione del driver più opportuno, quali la mancata corrispondenza di alcuni tra i comandi ed i set di caratteri della stampante. Spesso, ciò non consente di sfruttarne a pieno tutte le caratteristiche, poiché molti driver forniti dalla Commodore sono adatti a più modelli di stampanti.
Printer Driver Modifier (contenuto nella directory «PFM_Tools») analizza e permette di apportare modifiche al driver della stampante in seguito a problemi occorsi con la gestione dei font da parte della periferica.
Altra interessante utility è TextChars, la cui funzione è quella di ridurre al minimo i dati associati ad un font da trasferire (dal momento che non sono necessari tutti i caratteri presenti in un font), utilizzabile allorquando la memoria Ram della stampante sia troppo esigua e quindi insufficiente a contenere tutti i dati del set di caratteri.
Alcuni wordprocessor (tra cui «C1-Text») contengono l'utile opzione di controllo stampa il cui compito è quello di trasferire alla stampante un font definito. Nei casi però in cui tali opzioni non risultino presenti o tali dati si desiderano trasferire senza caricare un programma di elaborazione testi o lo stesso «PFM», può essere necessario utilizzare l'utility PrintRawFile che assolve proprio a questo compito, permettendo l'invio immediato di uno o più font alla stampante.
RevLin è invece una curiosa utility in grado di stampare un file ASCII in modo speculare, utilizzabile per stampare in rilievo su le stampanti che gestiscono ad esempio lucidi o diversi materiali come l'alluminio. AskAssign è la versione interrativa del comnado Assign dell'AmigaDos.

Le utility accessorie sono contenute nel secondo dischetto di «PFM».

gestione dei pennelli (porzioni cioè di font equivalenti ai brush di «DPaint»). Affinché queste operazioni siano accessibili (e quindi attivate dal programma) bisogna aver specificato un pennello tramite **Definizione pennello**, che abbiamo incontrato nella descrizione dei bottoni del menu laterale grafico di «PFM», o averlo caricato tramite la prima opzione di questo menu (**Leggere Pennello**). Il brush non deve eccedere le misure di 255x255 pixel.

Scrivere pennello memorizza il brush; **Capovolgimento orizzontale/verticale** capovolgere nelle due dimensioni il brush, mentre **Ruotare 90 gradi** si limita a girarlo parzialmente; **Rendere corsivo** serve per lo più per trasformare in corsivo un carattere, e quindi inclina il font verso destra; **Rendere negativo** inverte tutti i punti dell'immagine.

Dimensione contiene una serie di sotto-opzioni che si incaricano di gestire le misure del brush. **Libera** ridimensiona liberamente la dimensione del brush tramite l'estensione scelta con il mouse: **Metà/Metà orizzontale/Metà verticale/Doppio/Doppio orizzontale/Doppio verticale** non richiedono ulteriori spiegazioni dal momento che la denominazione spiega la rispettiva funzione sulle dimensioni del brush. L'ultima opzione di questo menu è **Impugnare**. Essa gestisce il punto in cui il pennello-brush deve essere «attaccato» al puntatore del mouse. Il brush può essere tenuto per i quattro angoli, e cioè negli angoli in **Alto a sinistra**, in **Alto a destra**, in **Basso a sinistra** o in **Basso a destra**. L'ultima presa riguarda il **Centro** del brush e costituisce la presa di default.

IL MENU MACRO

La progettazione e il disegno di un font implicano una gran mole di lavoro ripetitivo. Per far sì che questo venga svolto automaticamente dal programma senza tedio da parte dell'utente, «PFM» si serve di macro elaborate tramite l'uso di comandi testuali che rispecchiano le funzioni presenti nei vari menu. Una macro è costituita quindi da un programma che può essere associato ad un tasto (le lettere dalla A alla Z per un totale di 26 macro). Il programma componente la macro viene redatto in caratteri ASCII e può quindi essere realizzato tramite un qualsiasi editor ASCII («CygnusEd», «C1-Text», etc.) utilizzando una serie di 59 comandi.

Leggere macro apre un requester denominato **quadro di richiesta macro** con l'elenco di tutte le macro associate ai vari tasti: i tasti con le macro sono seguiti da un nome ed indicati in colore più scuro; subito dopo, un altro request ci permette la selezione del file.
Scrivere macro memorizza in un file una macro. **Iniziare registrazione file** richiede prima la lettera della tastiera alla quale associare la macro, poi attiva la registazione segnalata dal puntatore che oltre, alla freccia, contiene una grossa «M». A questo punto la macro risulterà costituita da tutte le operazioni che eseguiremo sequenzialmente sia tramite le opzioni dei vari menu sia tramite le operazioni associate ai bottoni situati lateralmente a sinistra della finestra principale. Selezionando **Terminare registrazione** si pone fine alla memorizzazione della macro (si noti che la «M» scompare dal puntatore). La macro così registrata può essere eseguita, salvata o esaminata.

Cancellare macro elimina le macro specificate dall'utente. **Esaminare macro** richiede inizialmente la macro da esaminare e la visualizza mostrando una sequenza di comandi di quattro lettere. La deselezione da mouse di **Codici** produce la visualizzazione della macro usando denominazioni estese di più facile

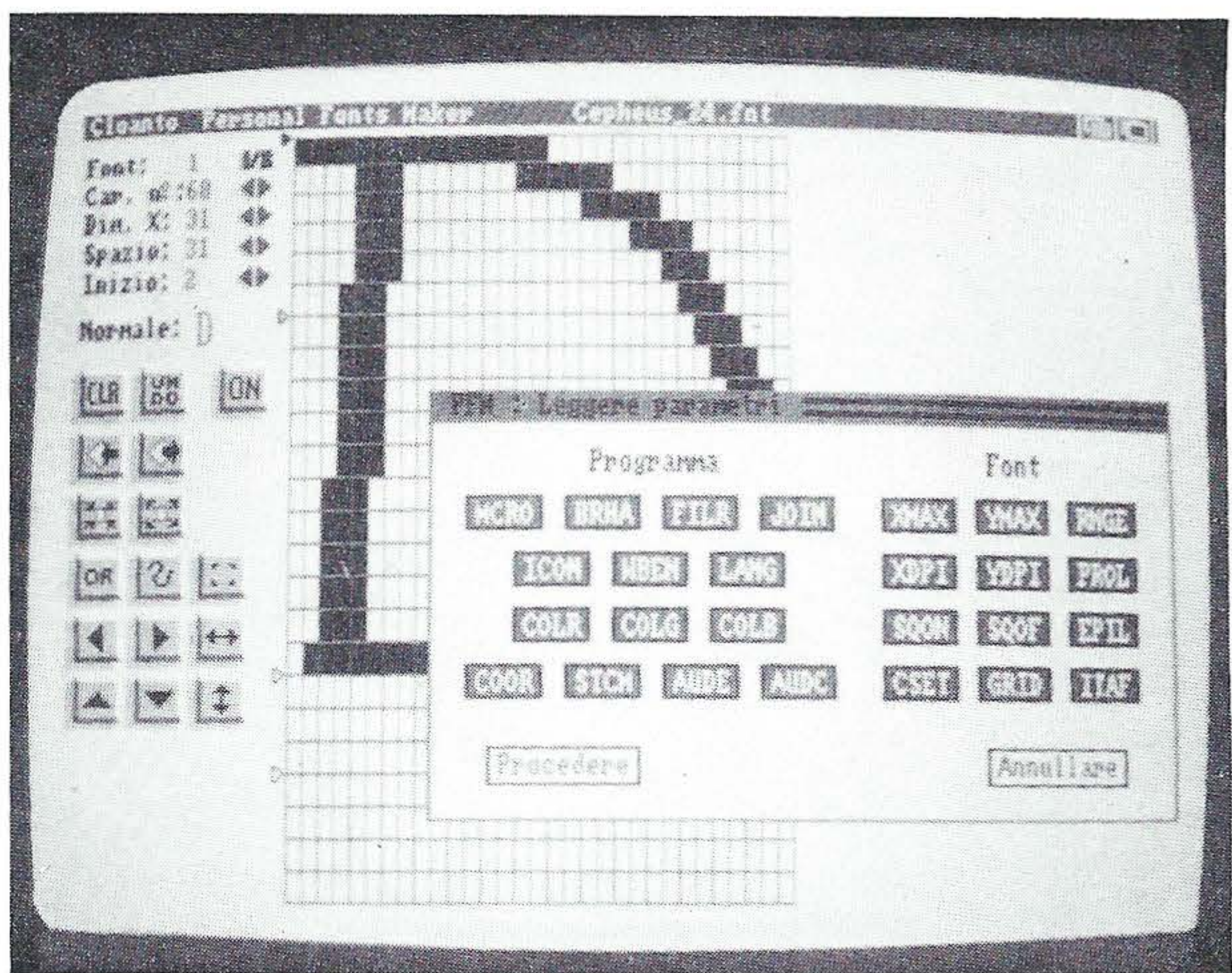



Fig. 6: tramite questo riquadro si attiva il filtro per l'attivazione dei parametri relativi al font ed all'ambiente di lavoro.

comprensione, mentre la selezione di **Numerare** esegue la numerazione delle linee macro.

Eseguire macro esegue una macro presente in memoria per intero se non è attivata **l'esecuzione controllata**, oppure una riga alla volta (il puntatore diviene una M lampeggiante) aspettando che l'utente prema il tasto Spazio per eseguire la successiva se essa è attivata. Il tasto Esc termina l'esecuzione. La title bar riporta il comando che sarà eseguito dopo la pressione del tasto Space.

I MENU PARAMETRI ED ATTRIBUTI

Il menu **Parametri** consente di adattare l'interfaccia del programma ai gusti dell'utente e fornisce alcuni parametri operativi.

«PFM» memorizza in un file tutte queste informazioni e può eventualmente ricaricarle in memoria tramite l'opzione **Leggere parametri**. Dopo la lettura del file, «PFM» presenta un quadro detto **filtro parametri** (Fig. 6) e, tramite la selezione di una o più voci, si può decidere quali dati attivare dopo la lettura. **Scrivere parametri** salva sia i parametri generali del programma che quelli attinenti ai font; anche in scrittura è possibile scegliere quali parametri salvare tramite l'apposito filtro.
Descrizione font mostra un quadro (Fig. 7) di parametri per l'invio di font alla stampante: **MAX X/MAX Y** specificano larghezza e lunghezza massima dei caratteri; **DPI X/Y** determinano la densità di stampa; **Prologo/Sequenze attivati/Sequenza non attivati/Epilogo** consentono di modificare le sequenze FFDL riguardanti il font corrente da inviare alla stampante.
Campo di iterazione contiene valori compresi tra 0 e 255 e stabilisce l'intervallo di caratteri (tra quelli attivati) da inviare alla stampante, o da memorizzare in un apposito file; **Fattore corsivo** chiede in input un valore che stabilisce l'angolo di inclinazione del pennello nell'opzione **Rendere corsivo**; **Coordinate** visualizza sulla title bar le coordinate del puntatore sulla griglia. L'utente può scegliere tra due sistemi di coordinate; la loro differenza si riduce al diverso inizio di numerazione del quadratino in alto a sinistra della griglia (**Origine 0: 0/Origine 1:1**). La pressione di un tasto del mouse produce la visualizzazione di coordinate relative, vale a dire che il punto in cui si preme il pulsante rappresenta l'origine del sistema di coordinate.
Griglia seleziona l'aspetto della griglia presente nella zona di editazione tramite le **sotto-opzioni Linee** e **Punti**, selezionabili anche separatamente. **Work Bench** permette di chiudere lo schermo del Work Bench per ridurre la memoria occupata, mentre **Icone** consente di scegliere se associare un'icona ai file salvati.

Dal momento che «PFM» permette di lavorare a due font contemporaneamente, **Agganciare font** fa sì che i due font siano direttamente confrontabili tramite il carattere corrente, che risulta invariato. **Modellare** adatta l'immagine di un solo carattere o di un intero font ad un diverso ambiente.
Tale adattamento può avvenire facendo sì che l'aspetto del font risulti immodificato (**X Proporzionale**); se invece non si vuole mantenere la proporzionalità tra le dimensioni x/y ma solo il rapporto tra la larghezza del carattere e la larghezza massima, occorre selezionare **X Massima**.
Richiesta file contiene quattro parametri per il controllo del modo di visualizzazione e selezione dei file dell'apposito request. **Lingua** non ha nessuna funzione nella versione italiana del programma, mentre in quelle per il mercato estero può selezionare i testi in diverse lingue.
Audio interviene sull'audio stereofonico associato dal programma in base alla selezione delle diverse funzioni.
Deviazione colori, infine, modifica tramite il famigliare request tutti i colori (otto) utilizzati dal programma.

L'ultimo menu di «PFM» elenca una serie di **Attributi** tipografici dal nome autoesplicativo (Corsivo, Neretto, Chiaro, Sottolineato, Contornato, Ombreggiato, Apice, Pedice, Espanso, Compresso, Negativo, Con grazie, Bozza, Passo fisso, Da destra, Dall'alto) che possono essere associati al font corrente.

Fig. 7: il riquadro di descrizione font consente di specificare i parametri per l'invio del font alla stampante.

TIRANDO LE SOMME...

«PFM» è un ottimo programma, unico nel suo genere; le operazioni consentite sui font sono sofisticate ma al contempo di facile utilizzo. La grande affidabilità del pacchetto è confermata dal fatto che, durante le prove, non abbiamo mai sperimentato crash del programma (leggi: visite del Guru...). Non c'è dubbio che «PFM» rappresenti attualmente una scelta praticamente obbligata per chi desideri creare e riprodurre font, su video e su carta, in maniera efficiente e veloce.

□

SupraDrive 500XP

Un hard disk per Amiga 500 che racchiude in poco spazio prestazioni ragguardevoli, facilità di installazione e 52 megabyte di capacità.

di VINCENZO MARANGONI

Il settore dell'informatica è tra i pochi in cui i prezzi dei prodotti tendono, con il passare del tempo, a diminuire invece che ad aumentare: accessori e computer che all'epoca della loro prima apparizione sul mercato costavano milioni ora sono disponibili a prezzi notevolmente inferiori, soprattutto grazie all'avvento di nuove tecnologie che hanno permesso di contenere i costi di produzione. Anche gli **hard disk** non fanno eccezione, ed in particolare quelli per Amiga: inizialmente considerati alla stegua di «beni di lusso» riservati solo agli utilizzatori professionali del computer, ora gli hard disk sono alla portata delle tasche della maggior parte degli utenti, anche di coloro che usano Amiga solo per giocare.

AGGIUNGERE UN HARD DISK

Se per Amiga 2000 erano già da tempo disponibili svariati modelli di hard disk dalle prestazioni (e dai prezzi) più disparate, è solo di recente che anche il fratello minore Amiga 500 è stato onorato dell'avvento di qualche modello di disco rigido ad esso dedicato. Il primo, ed ancora adesso il più diffuso, è stato il **modello A590** di produzione Commodore, un robusto e valido hard disk esterno da 20 Mb con un controller SCSI sul quale è possibile montare fino a 2 Mb di espansione di memoria ram.

L'A590 è tra gli accessori Commodore per Amiga più riusciti: il controller, pressoché identico al modello A2091 per Amiga 2000, consente di aggiungere facilmente altre periferiche aderenti allo standard SCSI (altri hard disk, Cd Rom, unità di backup a nastro, etc.), fino ad un massimo di sette, e può alloggiare sino a 16 chip di memoria, consentendo di aumentare la Ram del computer con un massimo di 2 mega in economia e senza occupare lo slot di espansione interno del computer.

Il Supra 500XP non necessita, solitamente, di un alimentatore esterno.

Il controller dell'A590 inoltre è autoconfigurante ed autobooting: la sua presenza (e la presenza di eventuale ram aggiuntiva ivi inserita) viene cioè automaticamente riconosciuta dall'hardware di Amiga, senza bisogno di routine software, e supporta la possibilità di eseguire da disco rigido il boot del computer.

Solo gli Amiga di produzione più recente, quelli dotati di KickStart 1.3, sono in grado di effettuare il boot (cioè l'avviamento del sistema) da hard disk, caricando da esso il WorkBench. Ai modelli più anziani, dotati di KickStart 1.2, può comunque essere collegato un disco rigido, ma il boot dovrà in ogni caso essere fatto inserendo nel floppy disk drive il dischetto WorkBench.

Per conoscere la versione di KickStart installata nel computer è sufficiente accenderlo e leggere il numero che appare nella schermata iniziale raffigurante la mano con il dischetto WorkBench. La versione del sistema operativo sugli Amiga più vecchi può comunque essere aggiornata senza troppa spesa semplicemente sostituendo un chip (operazione normalmente svolta dai centri di assistenza Commodore).

L'A590, nonostante i suoi numerosi pregi, presenta anche qualche inevitabile carenza, derivante dall'essere un prodotto relativamente a basso costo, sul mercato da un po' di tempo: le principali limitazioni consistono nella ridotta quantità di spazio a disposizione (un disco fisso da 20 Mega comincia ad essere insufficiente per chi utilizza attivamente Amiga, specialmente in ambito grafico o sonoro) e nella relativa lentezza di trasferimento dei dati.

L'ALTERNATIVA SUPRA

Per ovviare a queste lacune, numerose aziende produttrici di hardware per Amiga hanno introdotto sul mercato prodotti analoghi per Amiga 500, dalle prestazioni nella maggior parte dei casi superiori a quelle dell'hard disk Commodore. Tra esse spicca la statunitense Supra Corporation, con la sua serie di hard disk **SupraDrive** per Amiga 500 dotati, al pari dell'A590, di controller SCSI con espansione di memoria incorporata.

Il **modello 500XP** da noi provato montava un hard disk Quantum LP52S da 3"$^{1/2}$ avente la capacità formattata di 52 Mb (un formato che sta gradualmente sostituendo il tradizionale 40 Mb); il controller comprende un'espansione di memoria, inizialmente dotata di 512K di ram, che può alloggiare fino ad 8 mega di memoria aggiuntiva.

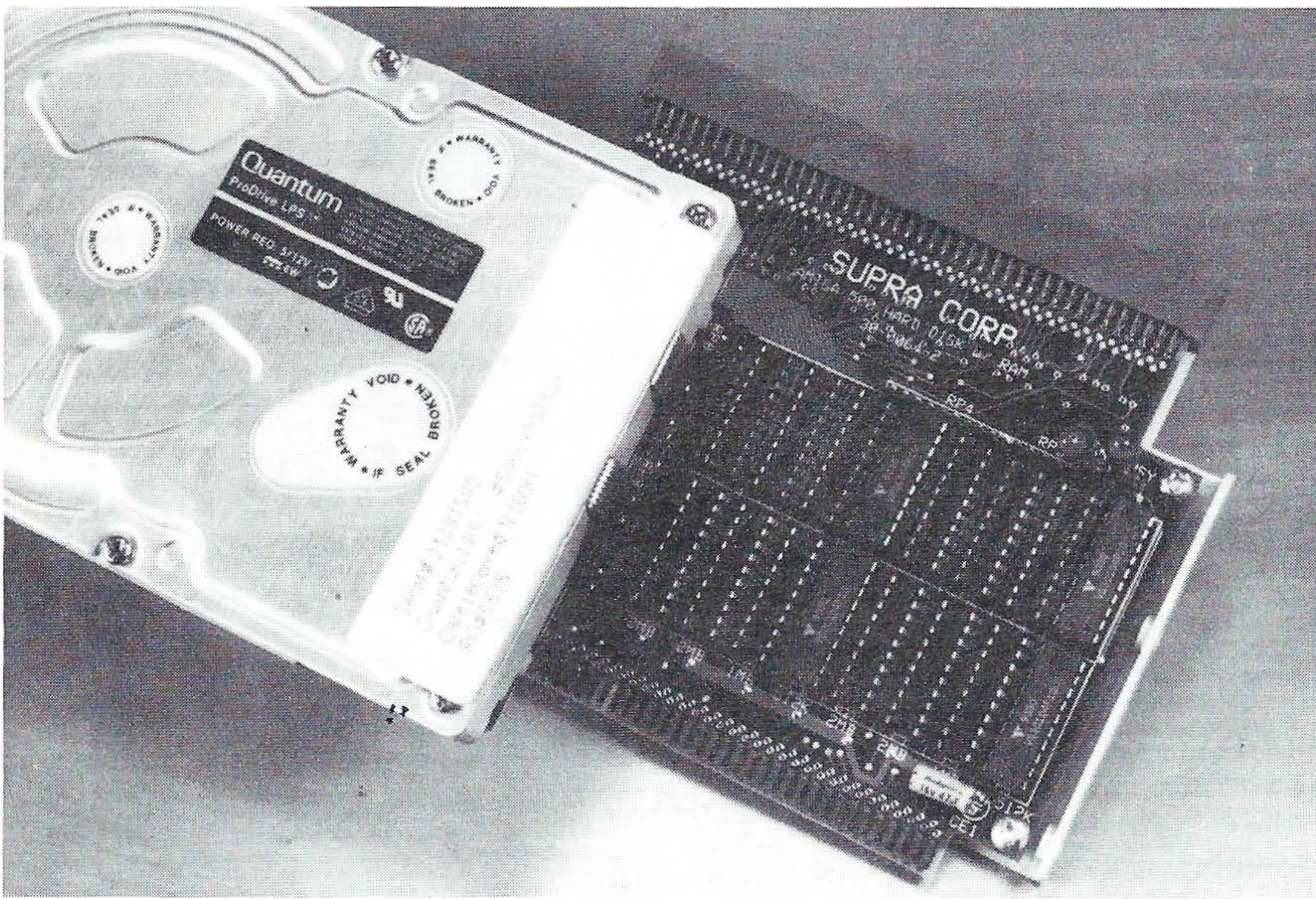
Come il A590, il 500XP si collega ad Amiga tramite il bus di espansione laterale ma, a differenza del concorrente Commodore, non impedisce l'aggiunta di altre periferiche in quello slot, essendo dotato di un connettore passante sul lato sinistro.

In più, il 500XP è dotato di un interruttore (denominato «**game switch**») che permette di disattivare l'hard disk e di renderlo invisibile ad Amiga senza bisogno di scollegarlo fisicamente: ciò risulta utile in quelle rare occasioni in cui sia necessario far girare programmi, nella maggior parte dei casi giochi, che non funzionano correttamente se è presente un hard disk collegato al sistema.

L'installazione del 500XP è realmente semplicissima e richiede solo pochi istanti. Venendo l'hard disk fornito già formattato e completo del software di sistema (il WorkBench), è sufficiente inserire il connettore a pressione nello slot laterale di Amiga (dopo aver naturalmente rimosso lo sportellino di plastica che lo ricopre) ed accendere il computer. Se Amiga è dotato di KickStart 1.3, il computer effettuerà il boot dal disco rigido e dopo pochi secondi ci si ritroverà nel cosueto ambiente WorkBench.

L'aspetto estetico del Supra XP 500 è gradevole, sebbene non particolarmente armonizzante con quello del computer, soprattutto per via del fatto che, una volta collegato, si trova in posizione asimmetrica rispetto ad esso (sporgendo di circa cinque centimetri sul fondo).

La prima cosa che si nota, rimosso l'apparecchio dalla confezione, è l'assenza di un alimentatore: il 500XP infatti è dotato di un drive «low power» a basso consumo di corrente e viene alimentato direttamente dal computer stesso attraverso il bus laterale. A questo proposito occorre segnalare che l'alimentatore di un Amiga 500, in condizioni normali, è predisposto per supportare il carico di tensione di al massimo due floppy disk drive esterni, e che quindi può avere problemi ad alimentare anche il 500XP (specialmente se viene installata al suo interno ulteriore memoria aggiuntiva) se vi è connesso più di un floppy disk drive.

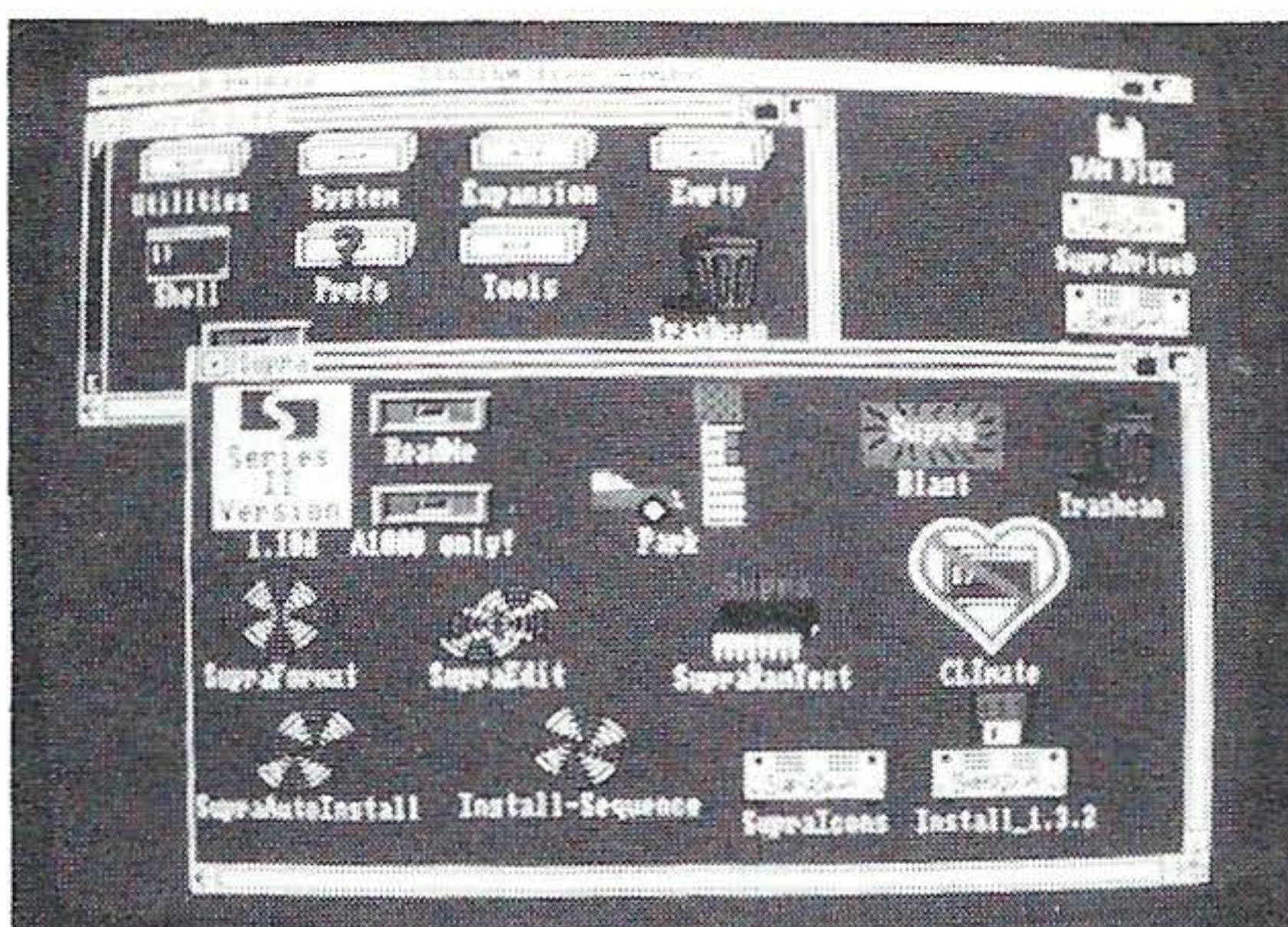
Abbiamo provato il 500XP, nella configurazione base con 512K di memoria, con un Amiga 500 dotato di un drive esterno e di 1 Mb di chip

Sulla scheda controller possono essere installati fino ad otto Mb di memoria fast da 120 ns.

ram, senza riscontrare alcun problema. Il manuale afferma comunque che, inserendo nella scheda più di 2 Mb di ram, occorre aggiungere un alimentatore esterno (che, per la cronaca, può essere lo stesso di un hard disk Commodore A590). In caso di alimentazione esterna, è indispensabile rimuovere preventivamente due jumper situati sul connettore dell'hard disk.

Per risparmiare memoria, è disponibile un'opzione di «auto-spegnimento» del disco rigido simile a quella implementata su parecchi computer portatili: tramite il comando «**SupraPDown**» (presente nella directory C: del disco) si può impostare un tempo di inattività, trascorso il quale l'hard disk si spegne automaticamente per risparmiare energia: qualsiasi accesso successivo al disco riaccenderà automaticamente l'unità. Lo svantaggio di questa funzione consiste nel fatto che la riaccensione richiede alcuni secondi e che, quindi, qualsiasi comando digitato dopo il periodo di spegnimento richiederà un tempo maggiore per essere portato a termine.

IL SOFTWARE

All'origine il disco da 52 Mb viene fornito già formattato e suddiviso in due partizioni (device) da 26 mega ciascuna, denominate **DH0:** e **DH1:**, che vengono riconosciute e configurate al momento del boot grazie al comando software «**SupraMount**» presente nella startup-sequence del disco. All'occorrenza questa configurazione può essere modificata a piacimento usando i programmi «**SupraFormat**» e «**SupraEdit**» forniti a corredo. A queste due utility, ricche di opzioni ed utilizzabili facilmente via mouse, si deve ricorrere qualora si desideri riformattare o ripartizionare il disco per esigenze particolari (ad esempio per creare una partizione da riservare per l'uso con emulatori MsDos o Macintosh): occorre evitare ovviamente di giocherellare con i vari gadget se non si è certi di quello che si sta facendo, visto che la variazione involontaria di qualche parametro può rendere inservibile il contenuto del disco.

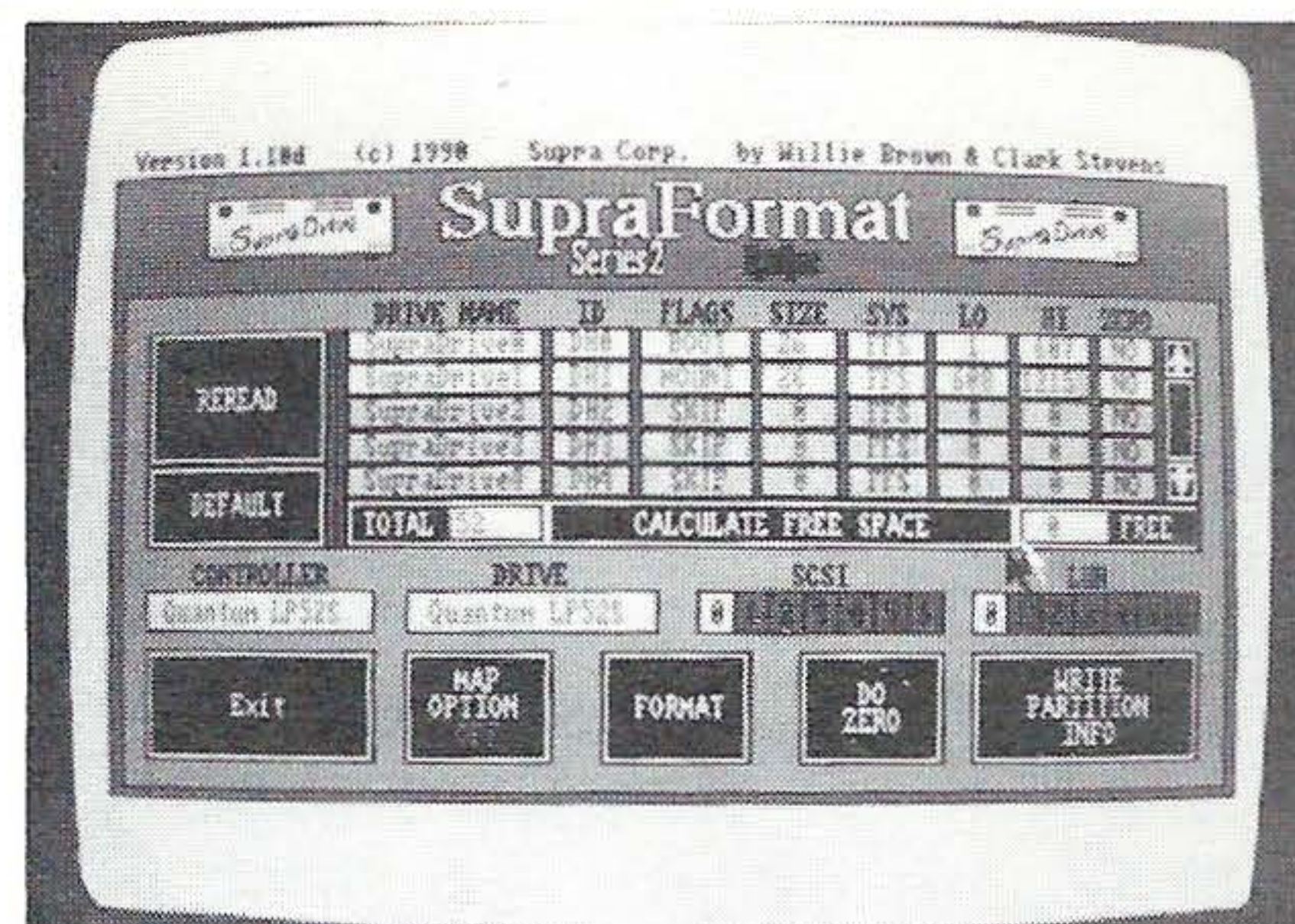
Il software a corredo è composto da tre dischetti; il contenuto dei primi due, quelli contenenti i tool sopracitati più altre utility tra le quali un programma per effettuare il test della ram ed il programma commerciale di gestione file «**CLImate**», è presente anche sull'hard disk stesso, che esce dalla fabbrica già formattato e pronto all'uso. I dischi vanno comunque conservati religiosamente poiché servono per reinstallare il software in caso di riformattazione.

Le utility in dotazione.

Il programma «Supra Format».

Il terzo disco comprende il programma «**ExpressCopy 1.17**» della Express Way Software, un'utility per effettuare il backup (copia di sicurezza) su floppy dei dati presenti sull'hard disk. Pur trattandosi di un bonus sicuramente gradito ai neo-acquirenti del 500XP, che potranno così disporre immediatamente di uno strumento in più per la salvaguardia dei loro dati, «ExpressCopy» è un programma, risalente al 1989, senza infamia e senza lode che non brilla particolarmente per velocità ed efficienza. Un buon investimento, più avanti nel tempo, consiste sicuramente nell'acquisto di un software di backup più avanzato e versatile (ad esempio «**QuarterBack 4.2**» della Central Coast Software).

La documentazione (purtroppo in inglese, almeno per il momento) è molto completa ed illustra chiaramente la (semplicissima) procedura di installazione, il significato degli switch posti sul lato dell'hard disk, ed il corretto metodo per aprire il coperchio ed inserire eventuali chip per espandere la memoria.

Gli switch sono cinque: i primi tre selezionano l'indirizzo del dispositivo SCSI del controller, il quarto attiva o disattiva la funzione di autobooting, ed il quinto viene usato solo quando si vuole effettuare il test della ram montata sulla scheda.

Fortunatamente il 500XP non richiede la rimozione del disco rigido per accedere agli zoccoli in cui inserire la ram (cosa necessaria invece con l'A590), e l'installazione dei chip addizionali risulta particolarmente semplice, a patto di stare attenti a non piegare i piedini e ad inserirli nel verso giusto. Una serie di jumper interni deve essere inoltre spostata in relazione alla quantità di ram aggiunta. Per espandere la memoria fino a 2 Mb è sufficiente aggiungere i normali chip di ram dinamiche da 256K x 4 (DRAM 414256 o equivalenti), che abbiano un tempo di accesso non superiore ai 120 nanosecondi. Espansioni superiori a 2 Mb richiedono invece l'impiego di chip da 1Mb x 4, più costosi e di reperibilità leggermente più difficoltosa. In linea generale, comunque, ogni 2 Mb di memoria in più vi costerà all'incirca 160.000 lire.

Le prestazioni del disco sono di tutto rispetto sotto il profilo della velocità. Dai test effettuati è risultato che, mentre il tempo di accesso medio ai dati è solo di poco inferiore a quello di un A590, l'incremento di velocità più ragguardevole si raggiunge sotto il profilo del «**transfer rate**» (cioè della velocità di trasferimento dei dati): questo valore è di circa 640Kb al secondo, ovvero poco più di tre volte quello di un A590. Ad onore del vero occorre precisare che il drive montato sugli A590 non è uno disco SCSI (generalmente i più veloci) al contrario di quello del Supra: la comparazione è quindi del tutto indicativa.

Il SupraDrive 500XP è disponibile in tre versioni, tutte dotate di 512K di memoria Ram, che si differenziano per la capacità del disco installato. I prezzi al pubblico sono i seguenti: 880.000 lire (con disco da 20 Mega), 1.200.000 lire (disco da 52 Mega) e 1.860.000 (disco da 100 Mega). La Supra Corporation produce inoltre versioni speciali per Amiga 1000 da 40 Mega (a 1.255.000 lire) e da 105 Mb (1.895.000 lire).

Si tratta di cifre proporzionalmente leggermente superiori a quelle di un A590, così come in proporzione sono superiori anche le prestazioni. I SupraDrive inoltre hanno una marcia in più: sono forniti con 512K di memoria già installata.

I prodotti Supra sono distribuiti da Flopperia, v.le Montenero 15, 20135 Milano. Tel. 02/55.18.04.84

Compilatore Benchmark Modula 2

Un compilatore Modula 2 che offre prestazioni di tutto rispetto, dotato di un ambiente di lavoro spartano ma confortevole e di un debugger veramente potente.

di PAOLO BOZZO

La volta scorsa abbiamo preso in considerazione le caratteristiche fondamentali del linguaggio Modula 2, esaminandone la struttura ed i comandi principali; ora vediamo come si presenta, all'atto pratico, una valida implementazione di questo linguaggio su Amiga: il «**Benchmark Modula 2**» di Leon Frænkel della software house Avant Garde. Le caratteristiche di base del nostro linguaggio (la leggibilità, la potenza, il rigore formale) evidentemente sono energici stimoli per i programmatori affinché realizzino pacchetti con compilatori integrati altrettanto potenti, rigorosi ed utilizzabili. Nel complesso il «Benchmark Modula 2» va proprio in questo senso; ovviamente si tratta di un vero strumento, che esclude fronzoli ed ambienti di lavoro inutilmente colorati; è inoltre caldamente consigliabile possedere una buona conoscenza dell'ambiente CLI (che comunque è indispensabile anche per linguaggi quali il C e l'Assembler).

Il «file requester» dell'editor è piuttosto spartano, ma indubbiamente efficiente.

Gli «eseguibili» che costituiscono l'ossatura di questo pacchetto sono un compilatore («**M2**» di 103 K), un linker («**M21k**» di soli 17 K) ed un editor («**M2ED**» di 95 K); insieme ad altri «attrezzi», essi sono contenuti appunto nella directory *Tools* (attrezzi) del primo disco, disco con il quale si può effettuare il *boot*, ovvero il lancio del sistema. Questo disco contiene anche, oltre che lo stretto indispensabile delle directory standard («c», «devs», «libs», etc.), una directory denominata **M2L**, alla quale la startup-sequence assegna il *device* logico M2L:, e che contiene una vasta libreria di funzioni che permettono di interfacciarsi con il sistema operativo di Amiga.

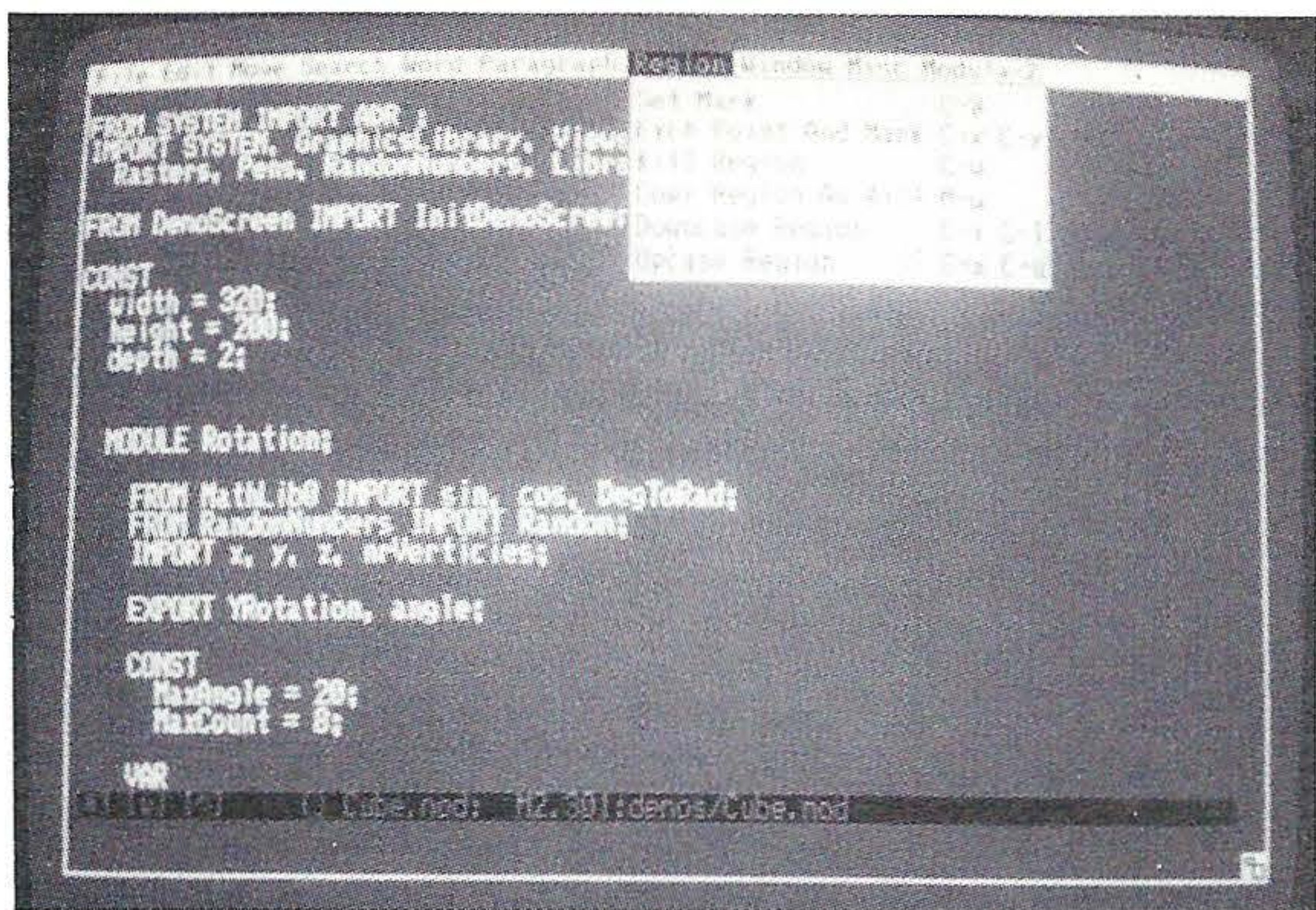
Il secondo disco è dedicato al **debugger**, vale a dire a quel preziosissimo strumento che permette di scovare gli errori che, come tanti piccoli *bug* (cioè insettucoli o pulci o «bachi»), si annidano nei programmi e che costituiscono un malcelato tormento esistenziale per la maggior parte dei programmatori. Anche questo disco è *boot-abile* (cioè si può usare per il lancio del sistema) e vi trova posto, oltre che il compilatore ed altre utility, il debugger vero e proprio («**M2Debug**», lungo ben 181 K).

L'ultimo disco contiene l'interfaccia per l'*assembler* e tre file compressi che, una volta scompattati, generano a loro volta tre dischi pieni zeppi di interessante *software* di pubblico dominio, naturalmente scritto in Modula 2.

NECESSITÀ DI MEMORIA

Chi possiede un Amiga 500 senza espansioni di memoria ed il solo drive interno, può utilizzare il pacchetto realizzando una copia del primo disco come disco di lavoro; potremmo chiamare questa copia «**M2–Work**», servendoci dell'apposito menu del Workbench; si dovrà poi avere l'avvertenza di cancellare (nella copia!) il file M2Ed con un comando CLI di questo tipo:

1 > delete M2–Work:tools/M2Ed

e di sostituirlo con l'*editor* più semplice e meno ingombrante in memoria, anche se privo del pratico selettore di

Il menu dell'editor ha una voce «region», che consente di agire sui blocchi di testo.

file; per fare questo basterà dare il comando:

1> rename M2–Work:tools/M2Ed.tiny M2–Work: tools/M2Ed

Questa configurazione potrà andar bene per iniziare e per costruire qualche programma particolarmente semplice; tuttavia, quando le esigenze si faranno appena più avanzate, incontreremo subito serie difficoltà. Si tenga presente che senza uno stack di almeno 10000 bytes (utilizzate per questo il comando «stack 10000» in CLI) la *Guru Meditation* sarà praticamente inevitabile, se chiamate il compilatore direttamente da CLI e non dall'*editor*. Inoltre, la sequenza di lancio presente nel primo disco chiude la finestra CLI dopo aver lanciato l'*editor*: in questo modo, se uscite da quest'ultimo vi troverete nell'impossibilità di far partire qualsivoglia programma. Per lo stesso motivo, anche dall'*editor* non potrete più usare i comandi CLI, salvo incrociare le dita e premere CTRL-Z per aprire un'altra finestra CLI, sperando che la memoria sia sufficiente. Se a questo quadro poco confortante si aggiunge che il compilatore dovrà caricare le librerie da disco, allungando i tempi di lavoro, si giungerà alla conclusione che un mega di memoria RAM e due drive costituiscono la configurazione minima per poter lavorare decentemente.

Se si opera in condizioni particolarmente favorevoli (magari con altre espansioni di memoria) è decisamente consigliabile copiare tutti i 184 (!) file che costituiscono la libreria disponibile nel disco **Ram:**. Poiché il *File System* che gestisce i floppy disk su Amiga è piuttosto lento, per sveltire l'operazione di copia è stato escogitato un sistema piuttosto ingegnoso ed efficiente; nella directory «**Tools**» esiste un comando denominato «**QLoad**» che consente di raggruppare tutti i nostri 184 moduli in un unico grande file (che può essere salvato su disco); con lo stesso comando si può compiere l'operazione inversa, vale a dire caricare l'unico file e trasferirlo in Ram: «spezzettandolo» contemporaneamente nei vari moduli di libreria; rispetto ad un normale comando «copy» l'incremento di velocità è veramente sensibile.

Ecco il completo ambiente di lavoro del debugger di sorgente.

Come abbiamo già detto nell'articolo precedente, i moduli di libreria vanno a coppie (implementazione/definizione); naturalmente questi moduli si devono presentare sotto forma di codici-oggetto, quindi prima di essere utilizzati devono essere compilati. Così, alle 92 coppie di moduli già compilati forniti dal pacchetto (e che ci risparmiano, come dice il manuale, di «reinventare ogni volta la ruota»), possiamo aggiungere le nostre coppie di moduli, che riutilizzeremo nei vari programmi, così come vuole l'impostazione del Modula 2.

LIBRERIE MODULARI

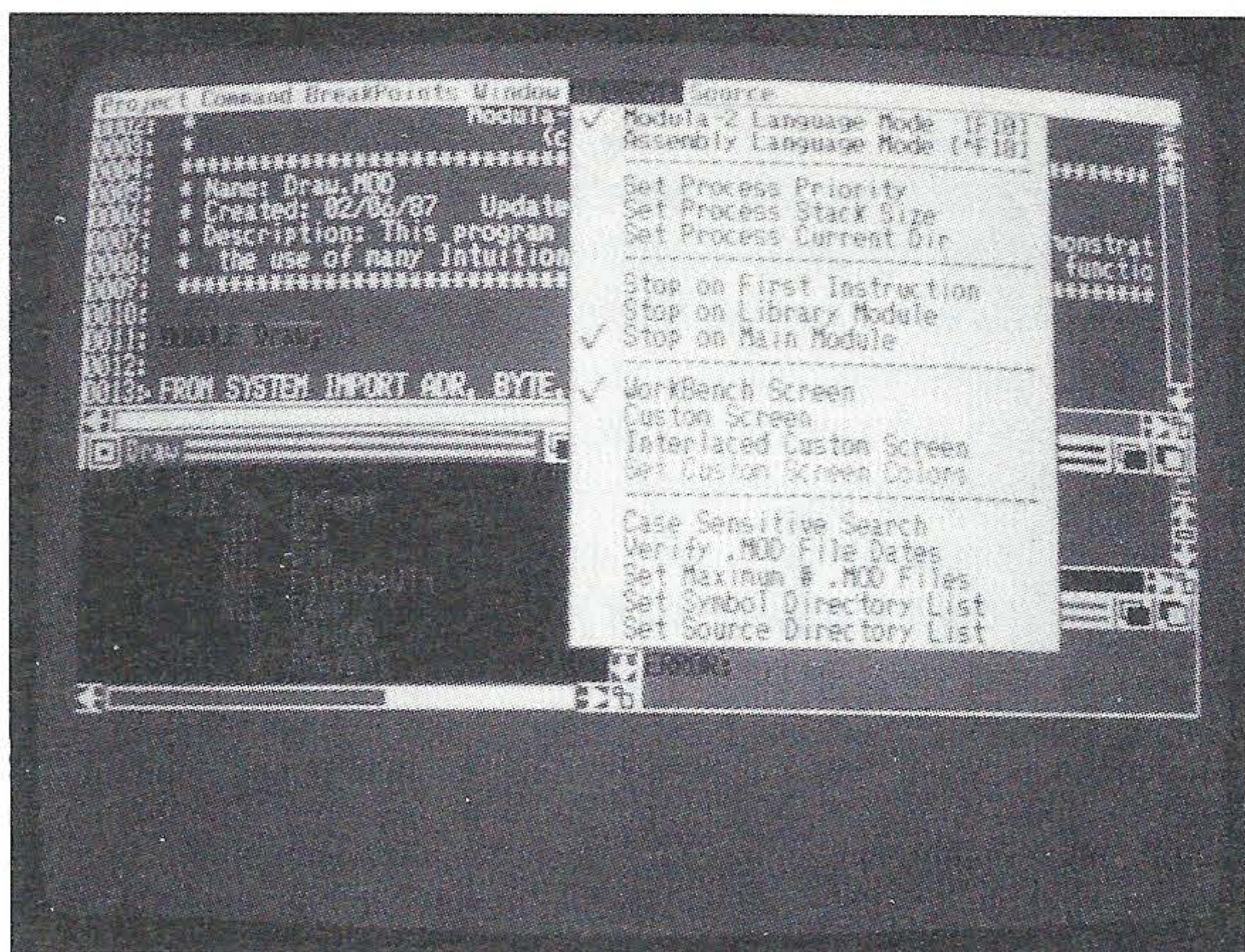
Come in ogni altro compilatore Modula 2, questi moduli di libreria sostituiscono sia i comandi **#include** al «preprocessore» del linguaggio C, sia le **librerie di linking** con estensione «.lib», come la ben nota «amiga.lib», utilizzabili attraverso i *linkers* standard «ALink» e «BLink». Ciò è possibile perché da questi moduli si possono importare, attraverso il comando **IMPORT** al quale abbiamo già accennato, ogni genere di oggetti: funzioni, procedure, costanti, record (equivalenti alle strutture C), etc. Si possono inoltre importare sia la definizione di un oggetto che la sua implementazione-dichiarazione.

La libreria è l'«anima» di un compilatore Modula 2 e quella fornita con questo pacchetto è esauriente. Gran parte del manuale d'uso, che è completo e sufficientemente pratico, è dedicata alla descrizione degli oggetti contenuti in questa libreria ed ai sorgenti di molti moduli-definizione; i sorgenti dei moduli-implementazione occuperebbero troppo spazio, e poi non sono indispensabili al programmatore, che ha bisogno di sapere solo come utilizzarli (senza contare che molti sono probabilmente scritti in linguaggio assembly).

Un grande pregio del Modula 2 è la portabilità, ovvero la possibilità di trasferire lo stesso codice-sorgente non solo su compilatori diversi ma anche su sistemi diversi. Per garantire questa portabilità esiste una raccolta di moduli di libreria standard che assicurano l'omogeneità con altri compilatori. Un esempio classico può essere costituito dal modulo **InOut** (abbreviazione di Input/Output, equivalente al file di include «stdio.h» del C), che contiene le procedure fondamentali di Input/Output come «WriteString» (per scrivere una stringa), «OpenInput», «OpenOuput» (per aprire file in input od ouput), e così via.

Se si vuole mantenere la possibilità di scrivere programmi portabili, magari verso sistemi Ms-Dos, è senz'altro conveniente utilizzare queste librerie standard. Purtroppo in questo caso bisognerà rinunciare all'interfaccia Intuition ad icone e menu, tipica di Amiga, il che rende assai meno attraente la possibilità di avere un codice portabile al 100%.

Per chi non ha interesse a queste operazioni, o le vuol mettere in secondo piano, esiste la possibilità di richiamare attraverso altri moduli le funzioni delle famose **shared library** (librerie condivise) di Amiga.

Il nostro compilatore ci permette anche, in diversi casi,

Numerose opzioni permettono al debugger di personalizzare il programma.

di «dimenticarci» delle chiamate alla funzione *OpenLibrary()* per aprire una libreria condivisa prima di usarne le funzioni. Non esiste però un sistema che consenta (come accade nel Lattice e nell'Aztec C, ed anche in qualche Basic) di porre i parametri di queste funzioni direttamente nei registri del 68000, guadagnando in dimensioni e velocità. La mancanza di un'opzione del genere può pesare anche nel caso si voglia utilizzare una libreria condivisa non creata dalla Commodore: in questa evenienza ci si dovrà rimboccare le maniche e realizzare delle coppie di moduli di libreria che fungano da «interfaccia» con il Modula 2; questo è il caso, per esempio, della «arp.library».

UN AMBIENTE COMPLETO E POTENTE

Una volta caricato l'editor ci troviamo in un ambiente integrato, completo e potente, nel quale possiamo digitare il testo nel nostro listato, ma anche compilarlo, collegarlo attraverso il *linker* ai moduli di libreria, e farlo girare.

Naturalmente quanto più è facile e manovrabile un editor, tanto più potremo concentrarci sul nostro lavoro. Al nostro editor «Emacs» non manca nulla: menu a tendina, finestre multiple, opzioni di ricerca... Un appunto però va fatto comunque, in quanto l'utente standard di Amiga non è abituato agli editor di testo di tipo «Emacs»: il programma «mEmacs», distribuito gratuitamente con la macchina sul dischetto «Extras», non ha mai avuto (ci sarà qualche motivo?) una grande diffusione ed un buon utilizzo; rischiamo dunque di trovarci un po' a disagio.

In primo luogo, le finestre multiple non sono finestre di Intuition, bensì solo partizioni orizzontali della finestra principale, e possono essere ridimensionate solo in senso verticale anche se, ad onor del vero, esistono tre gadget (individuati da [*], [v] e [^]) che permettono di chiudere o di dimensionare una di queste finestre anche con il mouse.

In secondo luogo, le varie voci del menu non solo introducono termini un po' strani per i normali utenti Amiga (*buffer, region, yank*) ma non sono neppure richiamabili con la classica combinazione di tasti che coinvolge il tasto Amiga destro, bensì con varie, ed a volte contorte, combinazioni fra i tasti Ctrl ed Esc, non sempre facili da ricordare. Queste combinazioni possono comunque essere ridefinite dall'utente, salvo Ctrl-G, usata per interrompere ogni operazione in corso.

Ci sono poi anche altre scelte che nascono in sintonia con la volontà di conformarsi allo standard «Emacs» (che, lo ricordiamo, è un editor utilizzato su grandi sistemi mainframe), come il particolare *display* delle stringhe che, per la loro lunghezza, andrebbero oltre il margine destro dello schermo. Ci pare insomma che il risultato sia quello di un certo disorientamento iniziale dell'utente non abituato ad altri sistemi né allo standard «Emacs»; insistendo nel lavoro, con il tempo, si scoprono comunque la completezza delle opzioni offerte ed anche la loro efficienza.

Il pacchetto offre, come esempio di «debugging», un semplice programma di disegno che può essere fatto girare completamente sotto il nostro controllo.

LA FINESTRA PRINCIPALE

La finestra principale è una finestra Intuition e ne ha tutte le caratteristiche; con l'opzione «**ToggleBorder**» del menu «**Winow**» è anche possibile aggiungere/togliere i bordi a questa finestra.

Come è noto, gli editor lavorano operando su dei *buffer* (zone riservate) di memoria, che possono essere riempiti di caratteri stampabili sullo schermo e che corrispondono ad un file su disco; «M2ed» può lavorare contemporaneamente su più di un buffer, anche se il buffer corrente è sempre solo uno. A questo scopo esiste un buffer speciale destinato a contenere la lista degli altri buffer.

Naturalmente si può operare anche su particolari zone di testo (cancellarle, spostarle, copiarle); queste aree, che gli altri programmi ci hanno abituato a individuare come «blocchi», in questo contesto vengono chiamate «regioni» e vanno manipolate in modo particolare. Per creare una regione bisogna «marcare» una posizione del cursore (esiste fortunatamente un modo molto semplice, che è il doppio click, ma si può usare anche il menu, oppure CTRL-@), quindi spostare il cursore nella posizione desiderata: la regione corrisponderà ai caratteri tra il punto marcato ed il cursore. L'operazione è semplice, ma si avrà — nei primi tempi — l'impressione che non abbia prodotto alcun risultato, perché la regione non verrà in alcun modo evidenziata; si rischia dunque di lavorare «al buio», e magari di cancellare ciò che non volevamo; l'unica verifica può venire dalla possibilità di invertire il cursore ed il punto marcato (due CTRL—x in sequenza).

All'editor non manca un selettore di file gestibile via mouse: si è dimostrato sempre piuttosto pratico ed intuitivo, e consente di visionare la lista dei volumi. È piuttosto veloce, ma purtroppo deve scorrere tutta la directory prima di lasciarci selezionare il file voluto e, come è noto,

I TASTI FUNZIONE DELL'EDITOR

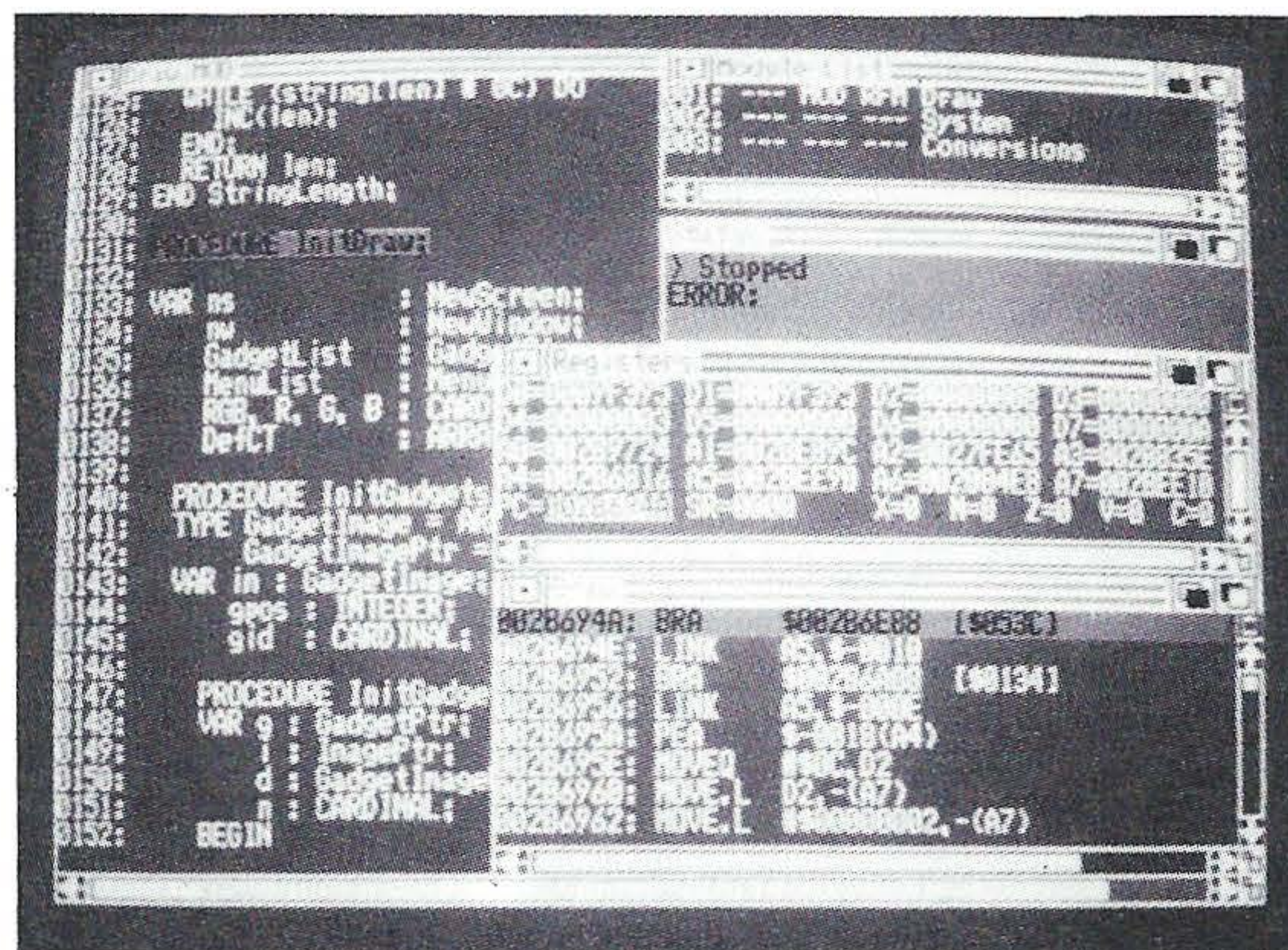
Tasto	Funzione
F1	**vai a errore successivo**
SHIFT+F1	**vai a errore precedente**
F2	**compila listato nel *buffer* corrente**
SHIFT+F2	**compila listato nel *buffer***
F3	***link* del modulo principale**
SHIFT+F3	***link* del modulo ...**
F4	**lancia il programma principale**
SHIFT+F4	**lancia il modulo ...**
F5	**compila un *file* da disco**
SHIFT+F5	**memoria libera**
F6	**seleziona il modulo principale**
SHIFT+F6	**seleziona directory uscita del *linker***
F7	**seleziona directory uscita del compilatore**
SHIFT+F7	**seleziona tabella dei simboli per *debugger***
F8	**carica il compilatore in memoria**
SHIFT+F8	**scarica il compilatore dalla memoria**
F9	**carica un file (*file requester*)**
SHIFT+F9	**carica un file in un'altra finestra**
F10	**salva il *buffer***
SHIFT+F10	**scrivi il file su disco**

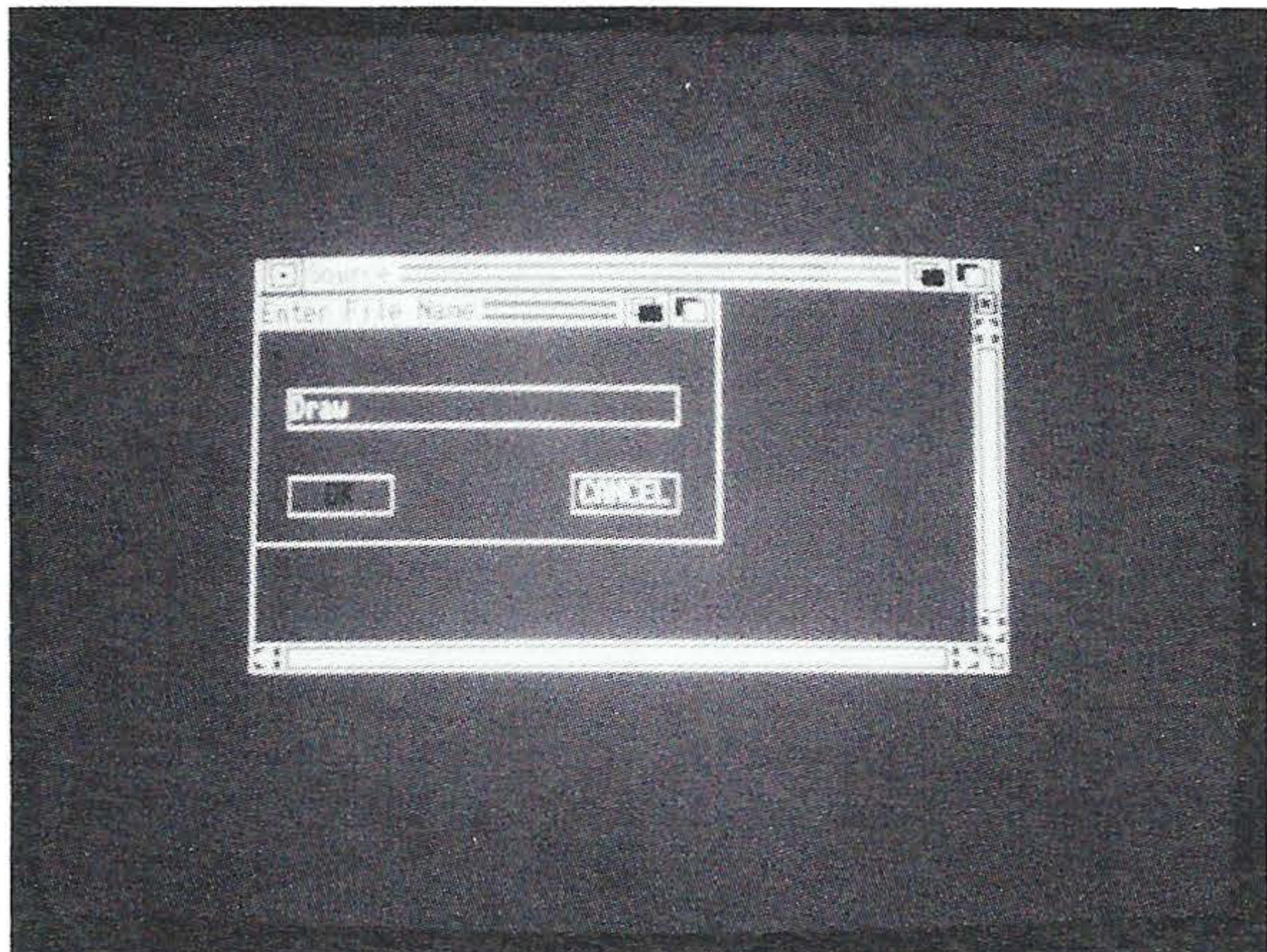
questa scelta può essere onerosa in termini di tempo quando non si opera con un hard disk e/o con il *Fast File System*. L'editor fornisce anche altre opportunità, quasi da Word Processor: esistono un menu «**Word**» ed un menu «**Paragraph**», con facoltà di spostarsi da una parola/paragrafo all'altra, di convertire in maiuscolo ed in minuscolo, di eliminare paragrafi o di spostarsi all'inizio o alla fine di essi. Un'altra caratteristica avanzata è data dalla possibilità di definire delle macro-istruzioni. Per definire una macro occorre: premere CTRL—x e successivamente «(» per segnalare che si inizia l'operazione; battere i tasti la cui esecuzione si vuol rendere automatica; infine, premere CTRL—x e successivamente «)» per segnalare che si termina l'operazione. Per richiamare la macro occorre premere CTRL—x seguito da «e».

Il menu che comunque solletica di più la nostra attenzione è denominato «**Modula—2**», e con esso si possono svolgere le seguenti operazioni: caricare/eliminare dalla memoria il compilatore; scegliere i parametri, lo *stack* e la finestra del programma da testare; definire le directory di ricerca/uscita del linker, etc. Queste ed altre opzioni di questo menu, nel loro complesso, rendono (piacevolmente) inutile servirsi dell'interfaccia CLI per le operazioni di compilazione, di linking, di prova del programma e di ricerca degli errori. Queste ultime operazioni si effettuano assai semplicemente ricorrendo ai tasti funzione, il cui uso abbiamo illustrato a parte.

UN DEBUGGER DI SORGENTE

Il vero fiore all'occhiello di questo compilatore è però il debugger «**M2Debug**», che consente di provare e di correggere un programma come, ed anzi meglio, di un interprete. Con questo debugger si può caricare in memoria un file eseguibile, generato dal compilatore «M2». Il file deve essere stato linkato con l'opzione (vedi riquadro) che inserisce nell'eseguibile i simboli necessari affinché il debugger possa riferirsi ai nomi degli oggetti utilizzati nel sorgente, ed orientarsi all'interno del sorgente stesso.

Una volta caricato in memoria l'eseguibile, ne verrà mostrato il codice sorgente Modula 2 e/o il disassemblato in comandi assembly per 68000. Si potrà quindi procedere all'esecuzione passo-passo (**step by step**) del programma, oppure si potranno fissare dei «punti di interruzione» (**break points**) in cui l'esecuzione verrà bloccata; potremo così prendere atto delle condizioni in cui si trovano il programma e le sue variabili in quel punto esatto, e procedere come sopra, sempre dietro nostro ordine; se si incontra o si crede di individuare qualche errore, si può tentare di correggerlo.

Un debugger veramente completo: ecco le finestre «register», con lo stato dei registri del 68000 e «Draw», con il disassemblato del codice Modula 2.

Come si vede, si tratta di un'esecuzione completamente sotto il nostro controllo, che ci consente di entrare nel programma in fase di esecuzione: uno strumento di sviluppo semplicemente eccezionale ed utilissimo. Proprio per evidenziare che si tratta di un programma a sé stante, al debugger è dedicato un manuale separato e completo. Questa volta ci troviamo di fronte ad un ambiente di lavoro perfettamente integrato con Amiga anzi bisogna ammettere che, facendo un confronto con l'analogo «**CodeProbe**» del compilatore C della SAS/Lattice, l'interfaccia Intuition e la facilità d'uso sono, nel nostro caso,

Purtroppo il debugger non ha un vero selettore di file; il requester qui evidenziato ricorda quello, tristemente noto, di AmigaBasic.

Il debugger consente di visionare record e variabili in modo straordinariamente completo e facile.

ancora migliori.

Il programma permette di aprire numerose finestre, ognuna con una propria funzione chiaramente identificabile, è dotato di un menu completo, ed è relativamente facile da utilizzare.

Per eseguire una linea di codice per volta basta premere F1: dopo che l'istruzione è stata eseguita, verrà evidenziata la linea seguente ed il debugger si metterà in attesa. Se si premerà F2 si otterrà un effetto simile, escludendo però dall'esecuzione passo-passo le procedure che verranno chiamate all'interno del nostro contesto. È un sistema molto pratico quando si sa già in quale procedura si annida il *bug*.

Premere F3 equivale al «trace mode» del Basic: le linee verranno eseguite senza attesa fra l'una e l'altra, ma verranno comunque evidenziate man mano che si procede. F4 è invece il corrispondente di F2, in quanto procede come F3 ma esclude dalla visualizzazione del sorgente il codice delle procedure che vengono chiamate.

USIAMO UN BREAK POINT

Il modo con cui si può agire sulle diverse finestre ha una sua coerenza interna, ed è basato sull'uso del mouse e sulla selezione dei vari oggetti, raffigurati simbolicamente all'interno della finestra. Per esempio, se vogliamo fissare un *break point* basterà attivare la finestra in cui appare il sorgente, selezionare con il mouse l'istruzione desiderata, ed effettuare un secondo click; un ulteriore click toglierà il *break point*. Una volta attivato un *break point*, possiamo scegliere «**Go**» dal menu «**Command**» oppure premere F6, ed il programma sarà eseguito velocemente fino al punto che avremo selezionato. A questo punto potremo controllare, per esempio, lo stato delle variabili, per vedere se il nostro codice ha ottenuto l'effetto desiderato: basterà spostarsi nella finestra «**Symbolic data**» e cercare il nome della variabile. Se invece che la variabile ci interessasse il campo di un record (equivalente al membro di una struttura in C), una volta individuatone il nome potremo letteralmente «entrarci dentro» selezionando il nome e poi facendo un altro click.

Le finestre di interfaccia sono di dieci tipi diversi:

1) source: contiene il sorgente in Modula 2, cui sono aggiunti i numeri di linea.
2) symbolic data: contiene le variabili, i record, etc., con i rispettivi valori.
3) memory: informa sullo stato della memoria.
4) disassembly: fornisce un disassemblatore completo ed utilissimo, specie nel caso in cui si abbiano routine in assembler.
5) registers: attraverso questa finestra si possono tenere sotto controllo i registri del 68000.
6) module list: contiene la lista dei moduli.
7) procedure call chain: concatenamento delle chiamate di procedura.
8) breakpoint list: lista dei break points.
9) info: dà informazioni sul sistema operativo, sulle risorse, sulle librerie, e così via.
10) directory: visualizza le directory ed il loro contenuto.

Nel complesso possono essere aperte più di dieci finestre, perché ogni finestra può essere aperta più di una volta. Ciò può tornare comodo in diversi casi, per esempio se vogliamo aver contemporaneamente visualizzate sullo schermo porzioni diverse di codice piuttosto distanti tra loro.

Naturalmente, anche questo ottimo strumento non ha possibilità illimitate, ed in alcuni casi si può arrivare al *crash* di sistema; si tratta però di limiti più che plausibili. Questo debugger risulta comunque molto potente e versatile, e completa in modo più che convincente un compilatore che, di per sé, è già promosso a pieni voti.

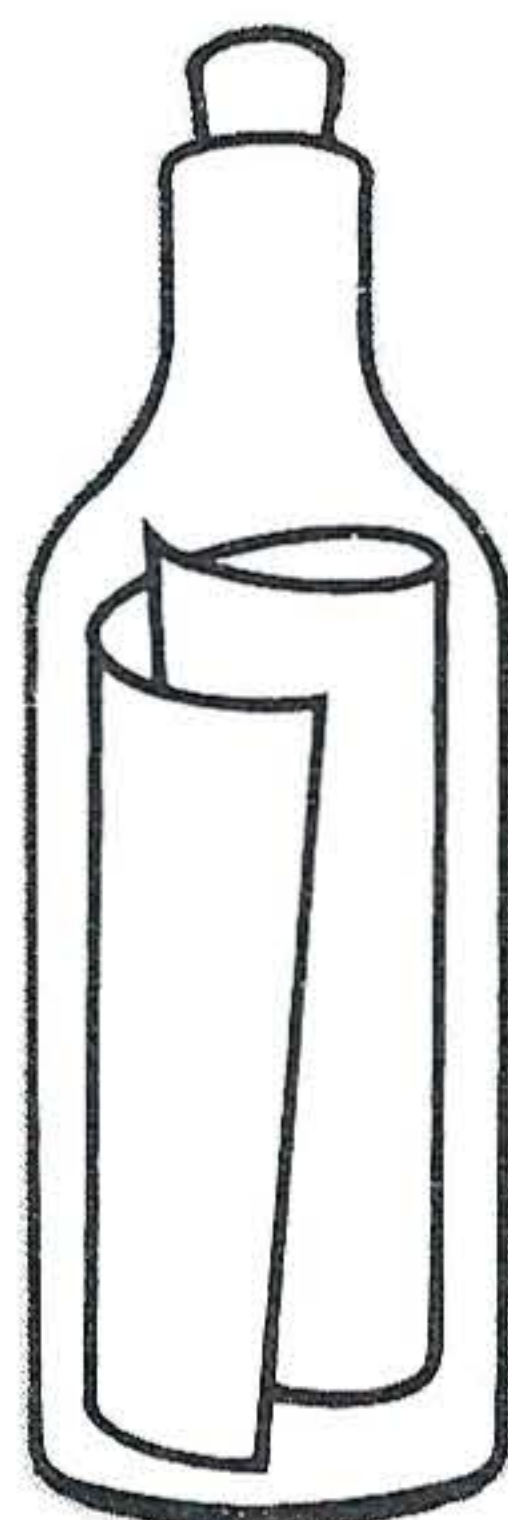
ASPETTANDO IL 2.0...

Ho saputo che sui prossimi modelli di A2000 ed A500 verrà montato in ROM il KickStart 2.0, e vorrei avere conferme in merito. Inoltre, c'è la possibilità che l'ECS sia presente sul prossimo A2000, permettendo l'eliminazione dello sfarfallio?

In questo caso che differenze sussistono tra un monitor MultiSync ed uno VGA? Vanno bene entrambi?

Marco Lorenzini - Muggiò (MI)

Per il momento, la Commodore non ha ancora annunciato ufficialmente l'intenzione di montare il Kickstart 2.0 su A2000 e A500, ma voci ufficiose assicurano che le Rom anche per questi modelli sono già in preparazione. Inoltre negli Stati Uniti è stata annunciata la commercializzazione di un nuovo Amiga 500P (dove la P sta per Professional), che dovrebbe comprendere di serie 1 mega di chip Ram, l'ECS e probabilmente anche il 2.0. Non resta che avere pazienza e vedere cosa ci riserva il futuro.

Riguardo all'ECS (Enhanced Chip Set), è necessario chiarire che esso non serve *ad eliminare lo sfarfallio, ma a permettere la gestione di un Mega di CHIP Ram, e la visualizzazione dei nuovi modi grafici denominati Super HiRes e Productivity. Per eliminare il flickering, è necessario un apposito circuito, come l'Amber (presente sulla scheda madre di Amiga 3000) o la scheda A2320 per Amiga 2000. In questo caso, è necessario un monitor VGA per visualizzare l'output del computer. Se si desidera avere la possibilità di disattivare il suddetto circuito, però, occorre un monitor capace di visualizzare sia i segnali VGA (31 Khz) sia il normale segnale Amiga (15 Khz). Un multisync di fascia bassa (o il bi-sync Commodore 1950) funzionerà egregiamente.*

I TEMPI POSTALI

È una settimana che ho inviato un vaglia per ricevere degli arretrati e ancora non ricevo nulla, come mai?

Luigi Di Stefano - Cosenza

I vaglia postali ordinari impiegano dai 15 ai 20 giorni ad arrivarci: da più lontano arrivano, più tardi arrivano! Consigliamo il vaglia telegrafico: scrivete però il vostro nome ed indirizzo di vostro pugno nella parte del vaglia che arriva a noi, altrimenti la trafila diviene ancora più lunga.

LE ISTRUZIONI COMPATTATE

Sono un vostro appassionato lettore da alcuni mesi e mi accingo a scrivervi per chiedervi come mai negli ultimi tre dischetti della rivista non riesco a stampare le istruzioni dei vari programmi o giochi, quando con le riviste precedenti non trovavo nessuna difficoltà a farlo, con qualsiasi programma word processor.

Antonio Villani - Nocera Inferiore (SA)

Le difficoltà riscontrate nel caricamento delle istruzioni dei programmi sono dovute al procedimento di compattazione, al quale sono sottoposte, allo scopo di aumentare lo spazio disponibile su disco. Il modo più semplice per ottenerne una copia su carta è comunque il seguente:
- effettuare il boot dal proprio dischetto Workbench (in quanto in quello di AmigaByte non sono presenti i driver per le stampanti);
- visualizzare le istruzioni secondo la consueta procedura (doppio click sull'icona corrispondente);
- durante la visualizzazione, premere i tasti «Shift», «Alt» e *«O».*

Il testo sarà così inviato alla stampante direttamente dal programma «Muchmore» utilizzato per la visualizzazione. In alternativa, le istruzioni possono essere scompattate tramite il programma «PowerPacker 2.3» (incluso nel dischetto allegato al numero 18 di AmigaByte) e riportate allo stato originario di file di testo Ascii, per poterle poi caricare con qualsiasi editor o word processor.

AMIGA BYTE LINE - 02/79.50.47 **HOT LINE!**

La Redazione risponde il mercoledì pomeriggio dalle 15 alle 18 a voce alle vostre telefonate.

BBS 2000 - 02/76.00.68.57 - 24 ore su 24

La Redazione risponde via modem nell'area «Linea diretta con AmigaByte». Collegatevi 300-1200-2400-9600-19200 baud.

RISERVATA AI LETTORI DI AMIGA BYTE

3D TEXT ANIMATOR 1.2

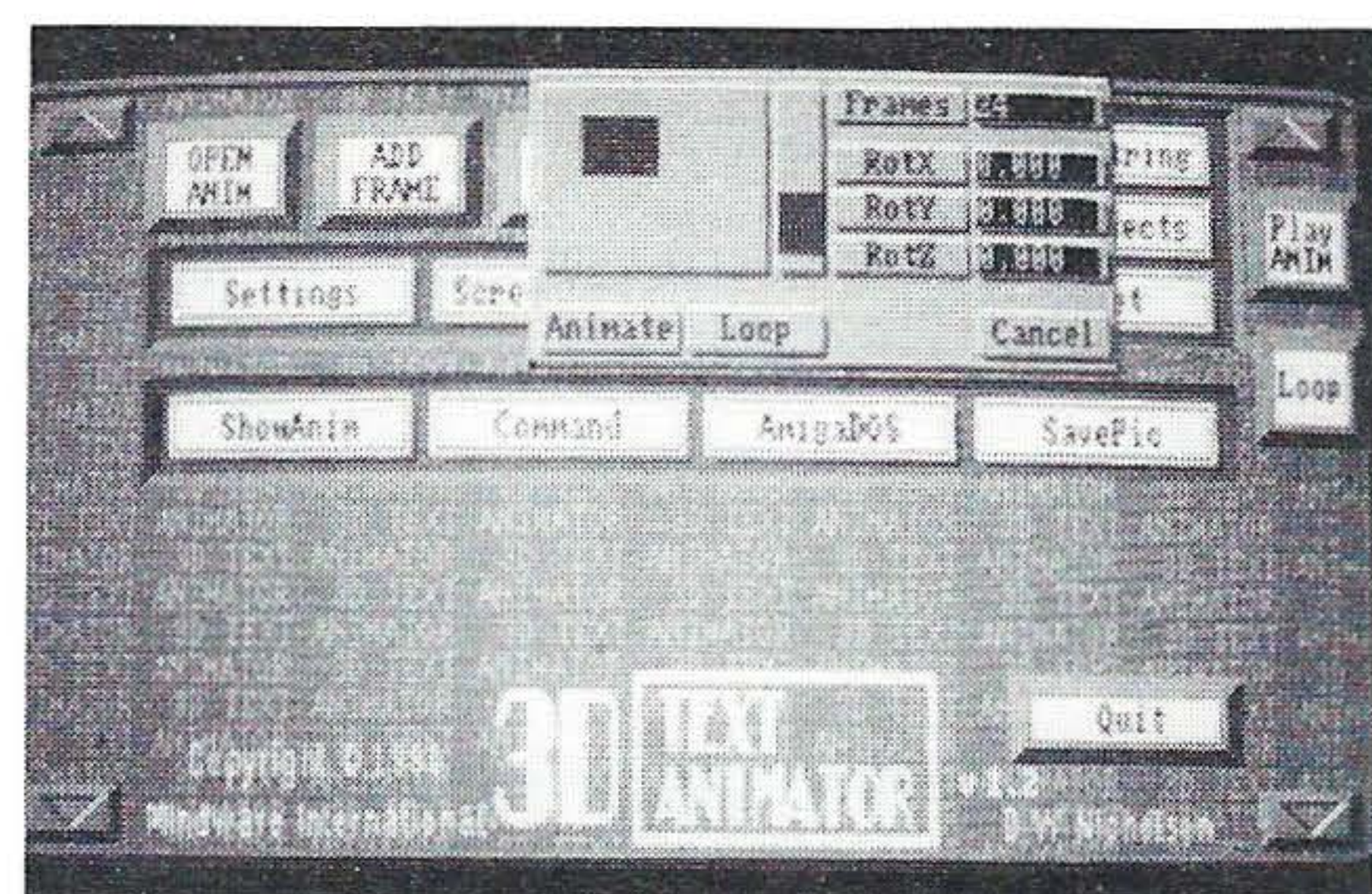
di Roberto Pellagatti

3D Text Animator è un'utility dedicata a coloro che utilizzano Amiga per la realizzazione di animazioni e di grafica nell'ambito del DeskTop Video o pubblicitario.

Come suggerisce il nome, «**3D Text**» (come definiremo d'ora il programma per brevità), consente di trasformare una semplice stringa di testo in un oggetto tridimensionale in movimento, salvando le animazioni nel formato standard Anim.

Una volta caricato il programma si nota immediatamente un'insolita interfaccia utente: l'ambiente di lavoro principale occupa solo la parte inferiore dello schermo e l'utente può interagire con esso tramite una serie di gadget. I tradizionali menu stile Intuition, al contrario, non vengono assolutamente utilizzati. Un secondo schermo, del quale è possibile scegliere la risoluzione grafica, permette di verificare l'effetto delle operazioni richieste. «3D Text» infatti non è un programma eseguibile vero e proprio, ma un'applicazione per il software «**PageRexx 3D**» della MindWare, progettata per essere eseguita in ambiente **TASS**.

Quest'ultimo programma (abbreviazione di «Thut Application Support System») è in pratica un'interfaccia utente semplificata, sviluppata dalla software house canadese Thut Inc. per poter utilizzare e lanciare applicazioni **ARexx** senza ricorrere al Cli.

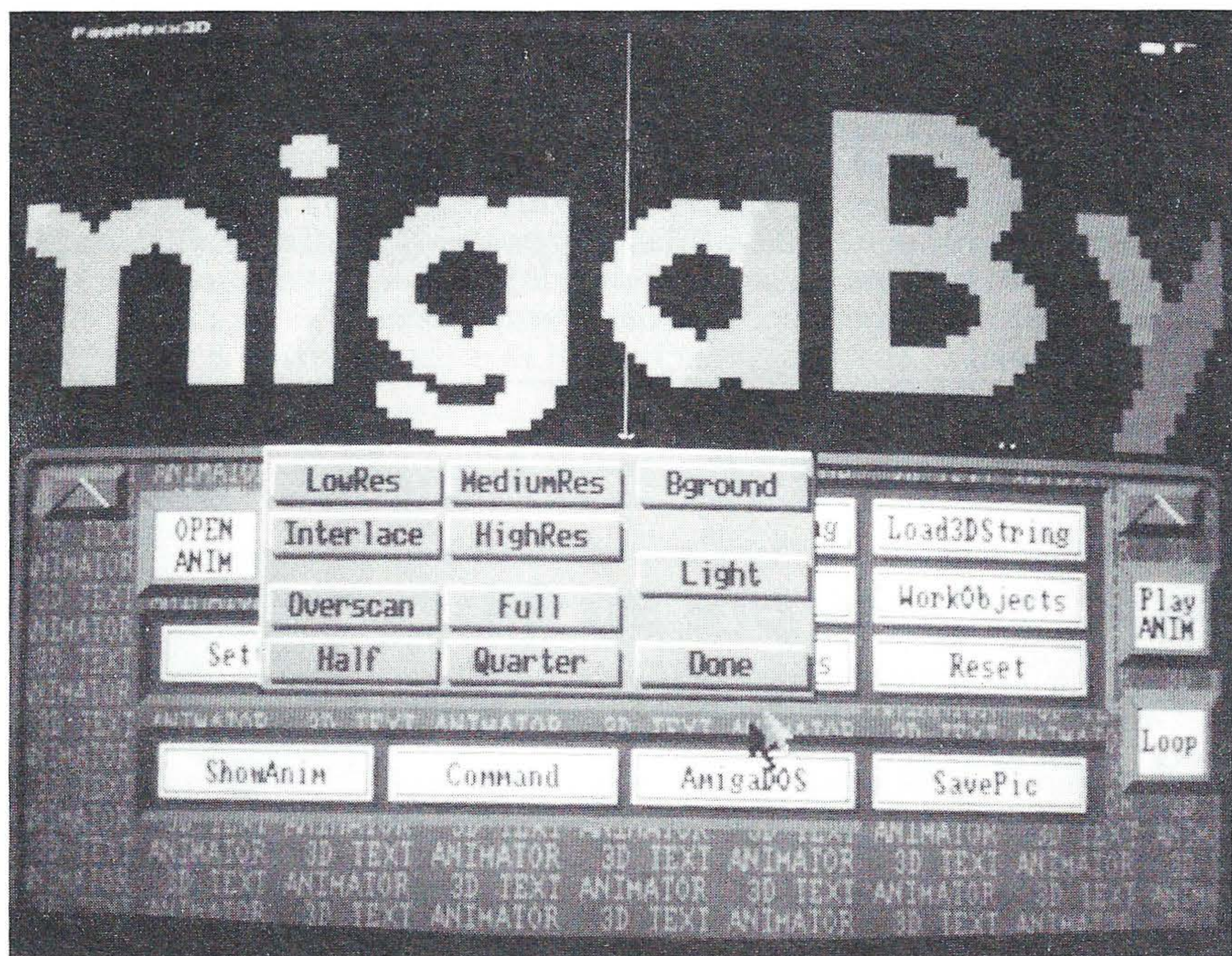
Tutti i programmi che possiedono una porta ARexx possono perciò essere pilotati, oltre che le macro scritte con quel linguaggio, anche tramite la più comoda interfaccia utente di TASS, che mette a disposizione un comodo sistema di requester, gadget, ed altre forme di input intuitivo.

«3D Text» quindi altro non è che una serie di macro TASS create per accedere, tramite ARexx, alle funzioni di «PageRexx 3D» (una versione utilizzabile solo tramite ARexx del programma di grafica «**Page Render 3D**», recensito da AmigaByte nel fascicolo 22). Lo schermo sullo sfondo di «3D Text» nel quale vengono tracciate le animazioni, infatti, riporta nella title bar il nome di «PageRexx 3». L'esecuzione di «3D Text» richiede perciò il previo caricamento dell'interprete TASS, incluso nel dischetto.

Creare una semplice scritta rotante è abbastanza semplice: una volta scelta la risoluzione grafica da usare con l'opzione **Setting**, si può specificare il testo da animare con i comandi **Load2Dstring** e **Load 3Dstring**, che sfruttano rispettivamente i normali font bidimensionali oppure quelli tridimensionali. Nel primo caso il programma provvede automaticamente a convertire in 3D i font bidimensionali mediante una semplice operazione di estrusione.

Con l'opzione **WorkObjects** si selezionano le lettere della stringa specificata che devono venire influenzate dai successivi comandi grafici di rotazione e traslazione.

Animate è invece il vero e proprio cuore del programma: selezionando questo gadget appare una finestra entro la quale è possibile specificare come la stringa debba essere ruotata in riferimento ai tre assi cartesiani.

Per quanto riguarda la definizione delle immagini create, «3D Text» prevede varie possibilità: la più semplice, in modo **WireFrame**, mostra solo i contorni dell'immagine senza colorarla, ed è adatta per un rapido controllo dei risultati con tempi di calcolo ridotti durante la creazione di un'animazione; la più sofisticata, in modo **Ray Tracing**, produce risultati molto più spettacolari, a scapito della velocità di elaborazione. A metà strada si colloca il modo **Dither**, che effettua una co-

lorazione veloce tramite retinatura, in maniera esteticamente accettabile.

Tornando alla gestione dei font, bisogna specificare che «3D Text» si comporta differentemente a seconda che la scrittura da animare sia composta con font 2D o 3D. Nel primo caso, ogni rotazione o spostamento influenzerà obbligatoriamente tutte le lettere componenti il testo, mentre nel secondo ogni lettera potrà essere selezionata ed animata singolarmente od insieme ad altre, per ottenere un più vasto numero di effetti.

Insieme a «3DText» vengono forniti due programmi separati, sempre da utilizzare in ambiente TASS: **ShowAnim** è un player di animazioni, mentre **Fed** permette di effettuare modifiche sui caratteri di un font e di trasformare un qualsiasi font bitmap bidimensionale di Amiga nel suo equivalente in 3D.

I creatori di «3D Text» hanno puntato molto alla completa compatibilità del loro software con gli altri programmi grafici disponibili per Amiga: la scelta del formato Anim, quasi universalmente adottato da tutti i pacchetti grafici, consente di effettuare un *merge* fra le animazioni create con «3D Text» e quelle ottenute, ad esempio, con «Deluxe Paint III».

«3D Text» non è certo un programma innovativo o particolarmente versatile: la varietà di animazioni realizzabili è piuttosto limitata dalla ridotta quantità di rotazioni possibili e dalla mancanza di effetti speciali.

Per questa ragione è indicato soprattutto ai principianti, o a coloro che non hanno necessità di realizzare animazioni particolarmente sofisticate e non vogliono avere a che fare con un programma con centinaia di opzioni che probabilmente non userebbero mai.

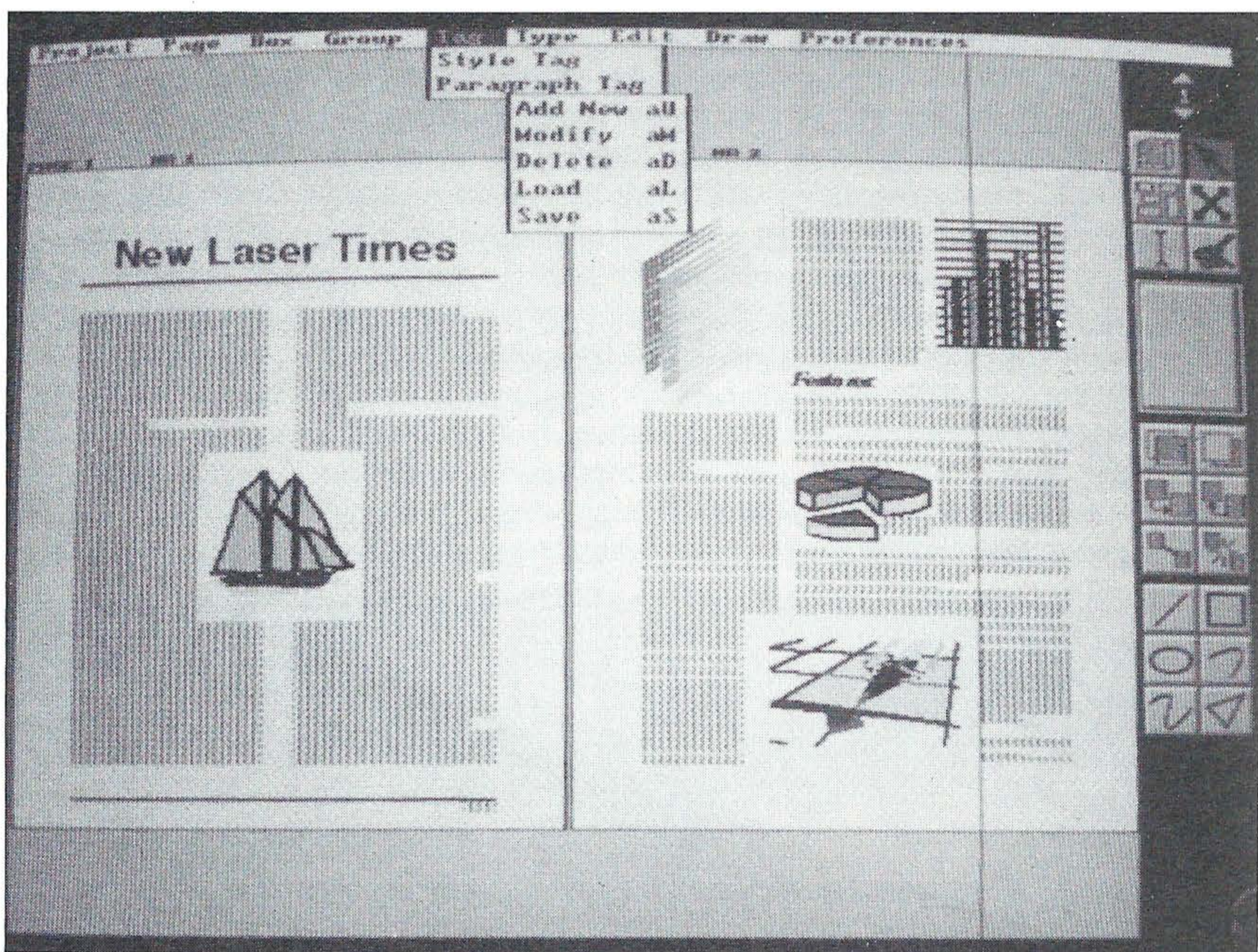
La strana interfaccia utente inoltre, per quanto effettivamente funzionale, può provocare più di una perplessità a chi è abituato ai tradizionali menu adottati dalla maggior parte del software Amiga.

La principale carenza di «3D Text» riguarda comunque la velocità: anche lavorando in modo Wire-Frame il programma è decisamente lento nel calcolare la rotazione delle stringhe, anche se composte da pochi caratteri, poiché ogni lettera viene rappresentata con una grande quantità di punti per permettere una buona resa dei dettagli.

Chi desiderasse eseguire elaborazioni molto complesse può comunque guidare il programma in modo automatico, tramite uno script ARexx, senza dover necessariamente seguire davanti al monitor tutte le fasi della realizzazione.

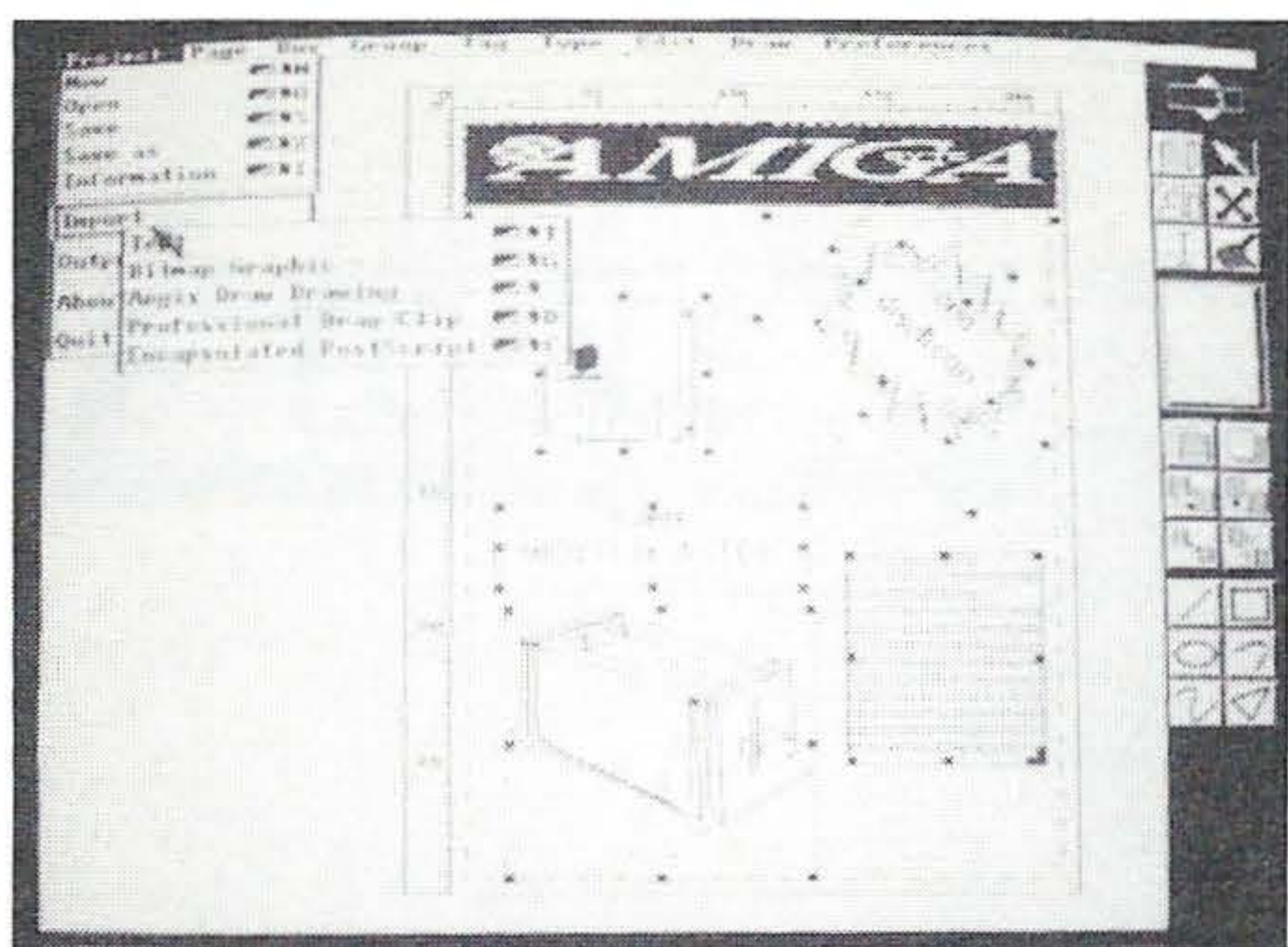
MINDWARE INTERNATIONAL
110 Dunlop Street W.
Box 22158, Barrie
Ontario L4M 5R3
CANADA

PROFESSIONAL PAGE 2.0

di Vittorio Ferraguti

Una volta caricata la nuova versione 2.0 di «**Professional Page**» («PPage 2.0» d'ora in poi) si ha l'impressione che ben poco sia in realtà cambiato rispetto alla precedente release 1.3, già recensita da AmigaByte nel fascicolo 19: gli stessi gadget, gli stessi menu, lo stesso look insomma.

Osservando meglio, tuttavia, le novità non tardano a farsi notare; prima tra tutti l'aggiunta di un nuovo menu chiamato **Tag**. Grazie ad esso è possibile definire una serie di parametri relativi agli stili del testo od al formato dei paragrafi, attribuendo loro un nome convenzionale richiamabile a piacimento.

Nell'ambito dello **Style Tag** si possono specificare il formato del paragrafo che si vuole utilizzare, il colore per il testo, il tipo di font, le sue dimensioni, il tipo di giustificazione, il kerning e lo stile (grassetto, italico, sottolineato, etc.). Entro il **Paragraph Tag** invece si possono scegliere i margini, il tipo di indentatura, le dimensioni delle tabulazioni e gli spazi fra le parole.

I Tag, una volta definiti, possono essere richiamati tramite il **menu Type** ed essere «applicati» sulla porzione di testo desiderata per attribuirle rapidamente le caratteristiche corrispondenti. Questa tecnica,

effettivamente molto comoda, equivale in pratica a quella introdotta per la prima volta nell'ambito del DTP su Amiga da «**Saxon Publisher**» (del quale è da poco in circolazione la nuova relase 1.1).

Un effetto collaterale causato dalla presenza di questi nuovi Tag, è che i documenti creati con «PPage 2.0» non possono essere caricati con la precedente versione 1.3, mentre è vero il contrario.

Un'ulteriore novità è rappresentata dal comodissimo **Article Editor**, che consiste in un vero e proprio word processor che opera in uno schermo di testo dedicato, invece che sulla pagina grafica di «PPage». Il principale vantaggio di questa innovazione è la notevole velocità di stesura e di correzione dei testi, decisamente maggiore di quella consentita dalle lentissime funzioni di input dei testi di «PPage 2.0» stesso.

L'editor, derivante dal programma «**Transcript**» della Gold Disk, include una quantità di opzioni davvero ragguardevole, tali da non far per nulla sentire la mancanza di un word processor WYSIWYG dedicato. Le uniche funzioni di cui è privo sono quelle relative all'impaginazione dei testi che, ovviamente, vengono svolte da «PPage 2.0» stesso.

L'Article Editor comprende, tra le altre cose, anche uno **spelling checker**, per controllare la corretta grafia di tutte le parole contenute nel testo sia in tempo reale (cioè man mano che vengono digitate) che a stesura completata; sfortunatamente il controllo viene effettuato in base ad un dizionario di termini inglesi e risulta quindi di scarsa utilità per un impaginatore nostrano.

L'editor consente l'import-export di documenti in formato «**WordPerfect**», e contiene una funzione denominata **Analyze Document** che mostra, sotto forma di grafico a barre, informazioni statistiche sul testo quali la frequenza delle parole basata sulla loro lunghezza, ed il numero di lettere, parole, frasi e paragrafi.

Con l'opzione **Send Text Home** infine si esce dall'Article Editor copiando automaticamente, all'interno del box originariamente creato con «PPage 2.0», tutto il testo digitato. Sebbene anche la precedente release 1.3 consentisse la creazione di pagine a colori, «PPage 2.0» è notevolmente potenziato sotto questo aspetto, ed esegue la separazione dei colori supportando ora anche direttamente lo standard tipografico **Pantone Matching System**.

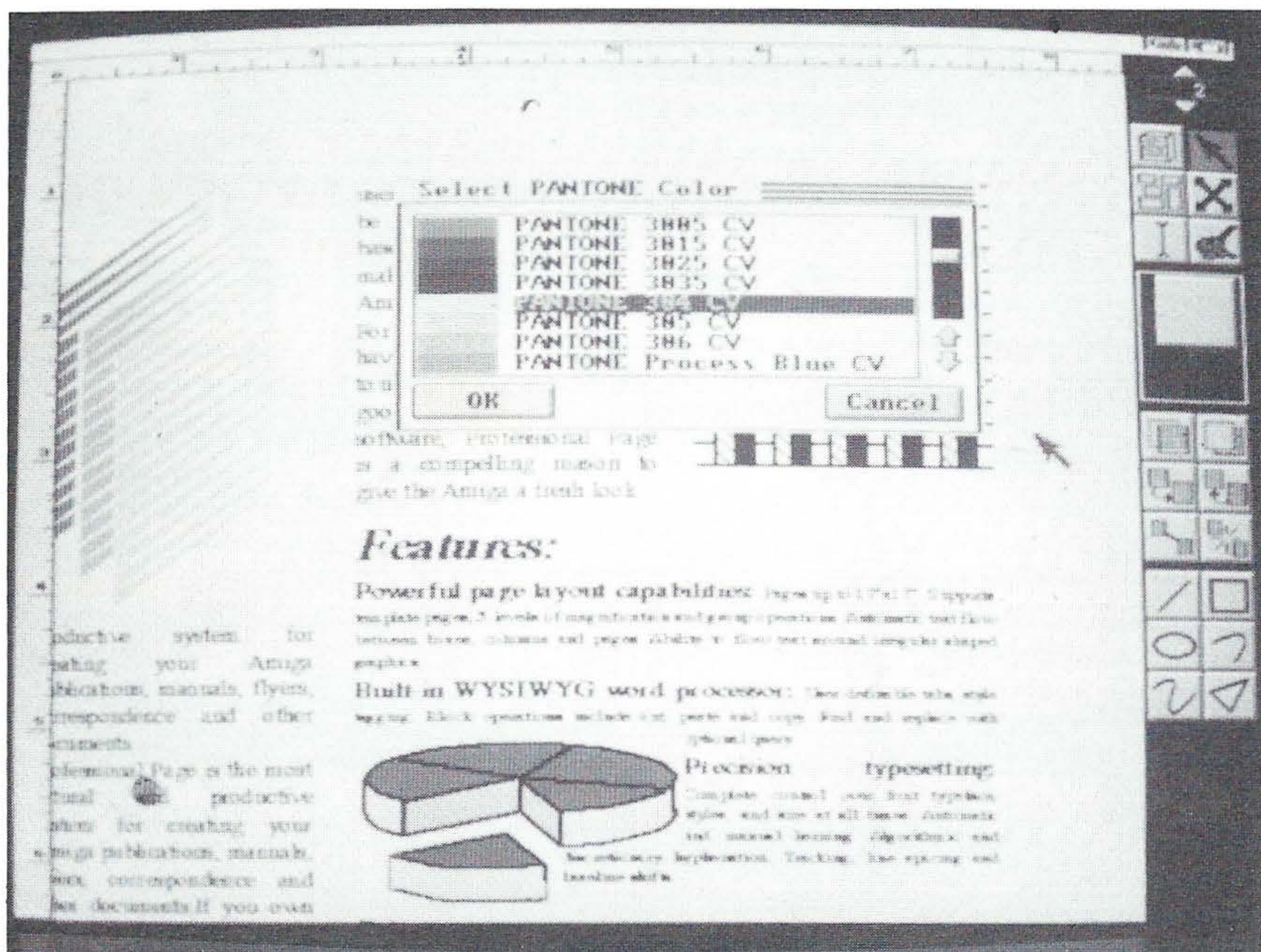
Sullo schermo può essere in teoria visualizzato contemporaneamente fino ad un migliaio di tinte diverse (ricorrendo alla tecnica del dithering) e, grazie al colore, anche i gadget della tool-box sono stati ridisegnati per conferire loro quel caratteristico aspetto «in rilievo» tipico del sistema operativo 2.0.

La disponibilità di molta memoria chip (almeno un mega) è comunque vitale per sfruttare decentemente le potezialità cromatiche di «PPage 2.0»: disponendo infatti di soli 512K di chip ram, risulta possibile operare in alta risoluzione solo in bianco e nero. Nei box si possono ora importare immagini grafiche a 24 bit (16.7 milioni di colori), oltre che i già consueti disegni strutturati in formato «**Professional Draw**».

Questi ultimi, a differenza delle immagini bitmap-Iff, possono essere ridotti ed ingranditi in qualsiasi scala, o ruotati di qualsiasi inclinazione senza perdita di definizione. La possibilità di effettuare rotazioni per box contenenti testo e grafica e di far scorrere il testo intorno alle immagini in maniera irregolare sono infatti altre due dimostrazioni della rinnovata versatilità di «PPage 2.0», che anche sotto questo profilo è ora in grado di competere con concorrenti quali «**PageStream**» o «Saxon Publisher».

Per quanto riguarda le altre caratteristiche del programma occorre precisare che, ancor più che nelle precedenti versioni, è necessario una quantità notevole di memoria per poter lavorare (almeno 2 mega); un hard disk e, possibilmente, una scheda acceleratrice sono altre due dotazioni hardware che il novello impaginatore con «PPage 2.0» dovrebbe seriamente prendere in considerazione (compatibilmente con le proprie finanze).

Oltre che per le prestazioni, «PPage 2.0» si fregia purtroppo dell'aggettivo «professionale» anche nel prezzo (circa 250$ negli USA) non proprio alla portata delle tasche dell'hobbysta medio. Ai possessori della versione 1.3 originale che decidessero di effettuare l'upgrade del programma e passare a «PPage 2.0», la Gold Disk regala un dischetto «**Upgrade Bonus**» contenente una serie di clip (disegni strutturati) in formato Pro Draw, ed alcune Templates (strutture predefinite di impaginazione) dimostrative, selezionate tra quelle disponibili commercialmente.

GOLD DISK INC.
5155 Spectrum Way, Unit 5
Missisauga, Ontario
L4W 5A1 - CANADA

Il fascino dei calcoli

Sposiamo grafica e matematica con «MathVision» e «Vista»: un connubio i cui frutti si rivelano gradevolissimi ed interessanti.

di LUCA MIRABELLI

Una suggestiva applicazione dell'apparentemente arida matematica è offerta da «**MathVision**», un pacchetto di programmi pubblicati dalla Seven Seas Software con lo scopo di creare suggestive immagini grafiche partendo da equazioni, formule e procedure. Se pensate ancora che il fascino di questo tipo di software non vi tocchi, date un'occhiata alle illustrazioni che accompagnano questo articolo.

«MathVision» è un programma molto versatile: gira su tutte le possibili configurazioni Amiga, dal 500 senza espansione al 2000 con schede acceleratrici, al 3000 con coprocessore matematico e sistema operativo 2.0. Per lavorare alle massime risoluzioni, tuttavia, un minimo di un Megabyte di RAM è caldamente consigliabile.

Il programma principale è fornito in due versioni, che si differenziano per le librerie matematiche utilizzate: solitamente quella da preferire è la «**MathVision.ieee**», che ha una precisione di 16 cifre significative.

Caricato il programma principale, compare l'**Edit Screen**, ovvero lo schermo nel quale inseriremo e modificheremo le nostre formule. Da un primo sguardo si nota subito qualcosa di familiare: il programma discende direttamente da «**Doug's Math Aquarium**», dal quale ha ereditato sia buona parte dell'interfaccia utente che l'intera filosofia di utilizzo.

Il nostro primo approccio con «MathVision» sarà naturalmente un grafico di funzione: per iniziare, selezioniamo **New** dal primo menu, per togliere di mezzo la schermata e la funzio-

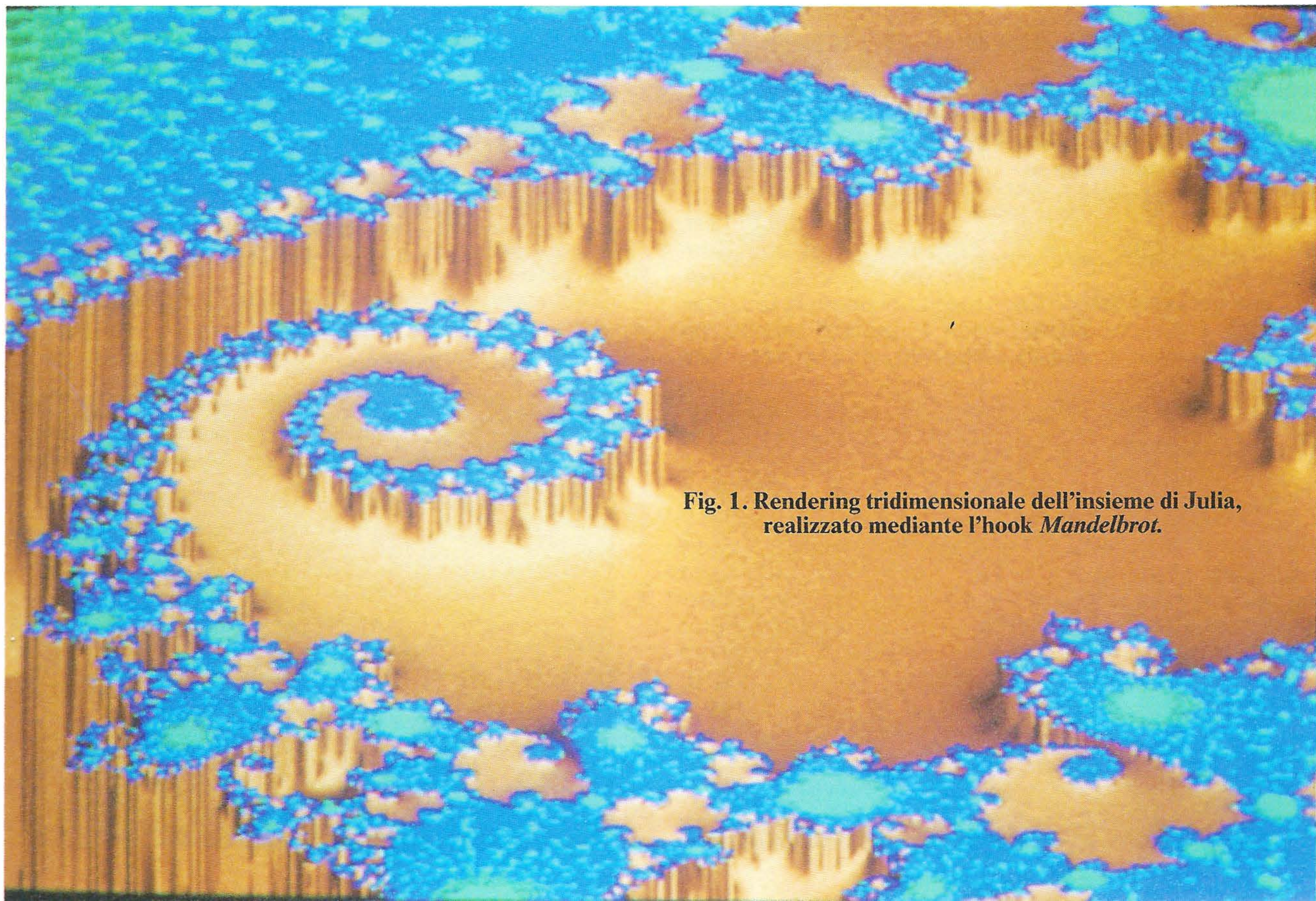



Fig. 1. Rendering tridimensionale dell'insieme di Julia, realizzato mediante l'hook *Mandelbrot*.

ne introduttiva. Presa per esempio la **funzione y=x*sin(x)**, scriviamone il secondo termine (la parte dopo il segno di uguale) nella **casella val**. Bene, ora ci rimane da decidere in quale intervallo studiarla: scegliamo **[−15,15]**, e riportiamo questi due valori rispettivamente nelle **caselle Xmin** e **Xmax**.

Dal **menu Simple** selezioniamo il comando **Guess Ymin Ymax** (che si occuperà di calcolare per noi il codominio approssimato della funzione), e finalmente **Plot**. Il risultato sarà il grafico desiderato.

Pronti? È ora di passare alle funzioni tridimensionali: questa volta la cavia sarà **z=(x*x+y*y)*sin(x*x+y*y)**.

Selezioniamo **New** dal primo menu e, come in precedenza, ricopiamo il secondo termine dell'equazione nella **casella val**. Questa volta, oltre che l'intervallo sull'asse X dovremo fornire anche un intervallo sull'asse Y: poniamo **Xmin** e **Ymin** a **−5**, e **Xmax** e **Ymax** a **+5**.

A questo punto è necessario compiere alcune operazioni sul **menu Perspective**. Selezioniamo per prima cosa **Guess Clip Limits**, seguito da **Sample Delta** (che porteremo ad un valore di **2**). Il **sottomenu Switches** è composto da una serie di voci che possono essere attivate o disattivate: disattiviamo **Color Top** e **Draw Box**.

Cosa abbiamo fatto? La curiosità è più che legittima: con il primo comando abbiamo incaricato il computer di calcolare il codominio approssimato della funzione (che compare nelle **caselle Clip High** e **Clip Low**); con il secondo abbiamo posto a 4 (2 elevato alla seconda) la definizione del grafico da tracciare, espressa naturalmente in pixel. Infine, abbiamo disattivato l'algoritmo di colorazione (vedi oltre), ed il tracciamento del parallelepipedo che contiene la funzione.

A questo punto selezioniamo **Plot** e, dopo un'attesa di durata variabile a seconda della nostra configurazione hardware, potremo ammirare sullo schermo il risultato dei nostri sforzi.

UN TOCCO DI COLORE

Attivando **Color Top** e **Color Bottom** possiamo colorare il grafico, attribuendo il colore ad un punto a seconda della sua quota. Ricopiamo i valori di **Clip High** e **Clip Low** nelle **caselle Contour Anchor** e **Contour Top**, avendo cura di premere <return> al termine di ciascun valore: questo farà sì che il parametro **Contour Width** (l'altezza delle fasce di colore) venga calcolato automaticamente.

Selezionando nuovamente **Perspective/Plot** otterremo il nuovo disegno.

Il terzo tipo di grafico supportato da «MathVision» è detto **Contour**, e consiste in una mappa bidimensionale della funzione, nella quale il colore di ogni punto dipende dal valore locale della funzione. In pratica, è come se guardassimo dall'alto l'ultimo grafico tracciato, quello tridimensionale con l'algoritmo di colorazione attivato.

Per tutti i tre tipi di plot, è presente il comando **Analyze** che consente di «passeggiare» lungo la funzione con il mouse, mentre un apposito riquadro fornisce le coordinate reali dei punti che tocchiamo, con informazioni circa eventuali punti di discontinuità.

Nel primo menu sono presenti i comandi per salvare il proprio lavoro su disco, in vari formati: solo immagine, immagine+formula, solo formula (il tutto in formato IFF), oppure come semplice file ASCII, per l'inclusione in testi o programmi.

Sotto la voce **Configuration**, inoltre, troviamo la possibilità di definire due parametri che hanno a che fare con il funzionamento del programma: il primo parametro è la precisione (ovvero il numero di cifre significative). Durante le prove effettuate, abbiamo osservato che la modifica della precisione non varia sensibilmente la velocità operativa del programma: una precisione bassa trova dunque la sua giustificazione solamente per effetti particolari desiderati dall'utente.

Anche la priorità può essere modificata, nel range −5/+5: in particolare, con valori minori di zero l'elaborazione avverrà negli «scarti» di CPU time avanzati dagli altri programmi in esecuzione, mentre con valori positivi «MathVision» avrà il sopravvento. Naturalmente, se non si sta lavorando in multitasking questo controllo non ha alcuna utilità pratica.

IL LATO ESTETICO

Abbiamo già osservato come «MathVision» sia un

UN ALTRO PUNTO DI... «VISTA»

La principale applicazione della geometria frattale è la simulazione del mondo naturale: il tracciato frastagliato di una catena montuosa, così come la forma delle foglie di felce, ben si presta ad essere calcolato con tecniche derivate da questa nuova scienza.

«Vista» è un degno rappresentante della categoria dei generatori di paesaggi immaginari. Partendo da una mappa (generata in modo pseudocasuale o caricata da disco), questo programma dà origine ad immagini piuttosto suggestive, quali il laghetto di montagna mostrato in figura. I controlli sono molto semplificati, ma non per questo poco potenti: con pochi click del mouse, in modo del tutto intuitivo, è possibile aggiungere laghi e fiumi (pulsanti *Lake* e *River*). Analogamente si posizionano l'immaginaria telecamera con la quale la scena dovrà essere ripresa, ed il punto al quale essa mira (*Camera* e *Target*).

I valori riportati nelle caselle *Tree* e *Snow* sono le quote alle quali, rispettivamente, cessano di esistere gli alberi e compaiono le nevi perenni.

Il pulsante *Haze*, se attivato, introduce un fattore di disturbo nella scena, una specie di nebbia che confonde i contorni delle cose. L'ammontare di questo effetto è specificato nella casella a fianco.

Blend aggiusta i colori nell'immagine finale in modo da

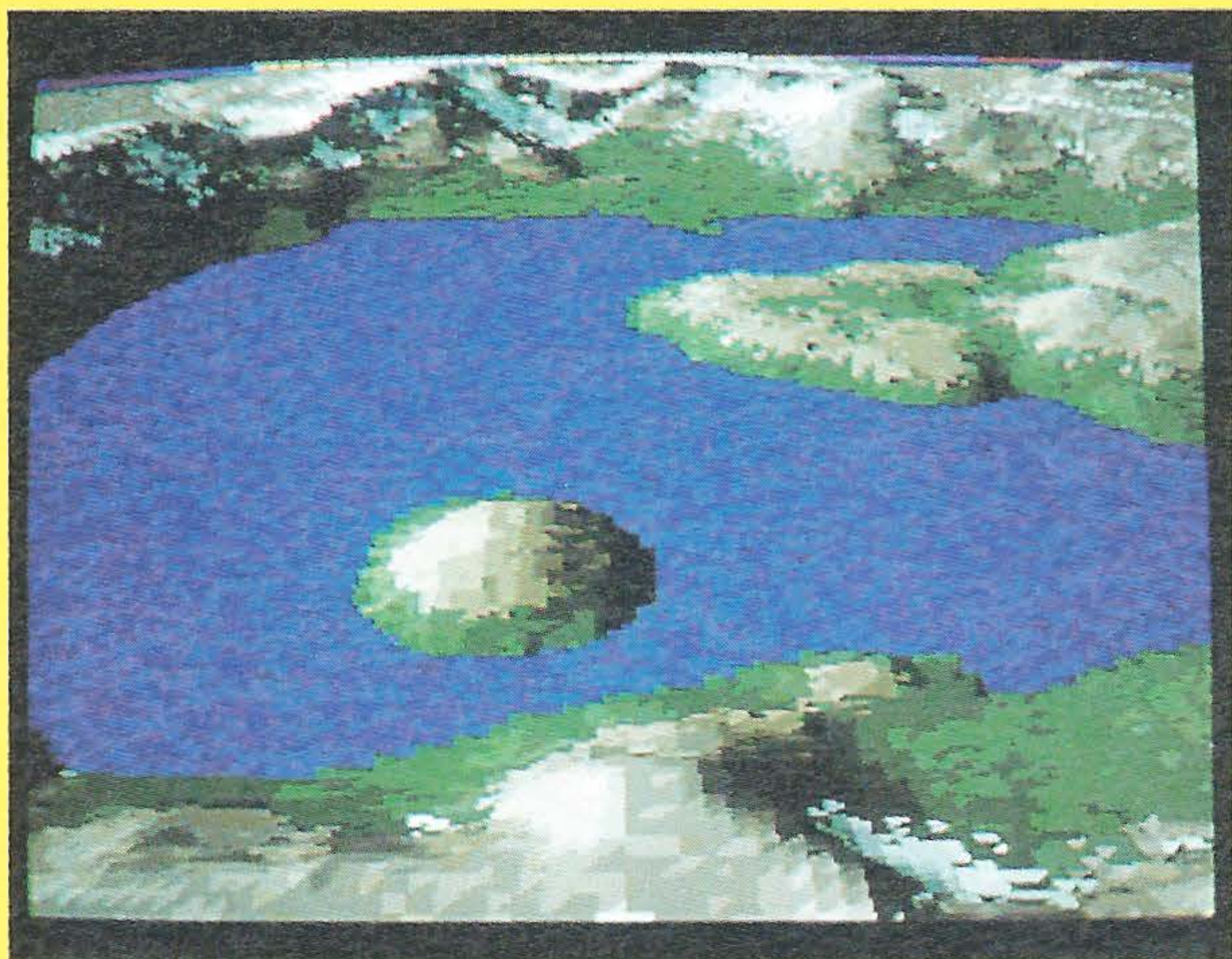
Così si presenta l'ambiente di lavoro principale di «Vista».

Fig. 2. In questo grafico, il colore di ciascun punto è derivato direttamente dal valore della funzione.

programma con una spiccata attitudine per la grafica: nel **menu Screen** si trovano i comandi per accedere al riquadro palette, per attivare o modificare il color cycling, e per settare le dimensioni dello schermo sul quale lavorare. In particolare, è possibile lavorare in tutti i modi grafici di un Amiga senza ECS, compresi Overscan, HAM e Halfbrite.

Con l'ausilio dei comandi del **menu Global**, invece, si può scegliere il numero di colori da utilizzare per il disegno (con **Modulo**), o tracciare un grafico senza cancellare lo schermo, cioè mantenendo come sfondo il grafico precedente, eventualmente sostituendo il colore zero con il trasparente (**Use Background**).

È giunto il momento di dare uno sguardo un po' più approfondito ai meccanismi di interpretazione della formula. Come abbiamo visto, l'espressione che fornisce il valore della funzione viene inserita nella **casella val**. Le altre, dalla **FA** alla **FI**, sono sotto-espressioni, utili per semplificare la comprensione della formula. Ad esempio, la ben nota funzione **z=(x*x+y*y)*sin(x*x+y*y)** può essere scritta come **fa*sin(fa)**, avendo cura di porre in **FA** la sotto-espressione **x*x+y*y**.

È possibile chiamare le sotto-espressioni con dei parametri (non c'è limite al loro numero); nella scrittura delle sotto-espressioni ci

Un laghetto di montagna, ottenuto con la massima facilità mediante l'opzione Lake.

evitare contrasti troppo rudi, mentre *Smooth* agisce a livello «fisico»: attutisce, cioè, le asperità del terreno. Mediante uno dei quattro pulsanti W, N, E, S si decide da quale direzione debba provenire la luce, mentre gli altri quattro, di fianco alla scritta *Poly*, selezionano il grado di precisione dell'immagine finita: il massimo grado è 1, che corrisponde anche alla massima durata dell'elaborazione.

***Wide* e *Zoom* sono i due tipi di obiettivo che è possibile utilizzare; *Render* calcola l'intera scena, mentre *View* mostra l'ultima calcolata.**

Il valore numerico dell'ultima casella in basso è il «seme» per la generazione di numeri pseudocasuali: non occorre preoccuparsi granché del suo significato, basta tenere presente che allo stesso numero corrisponde sempre la stessa mappa. Il gadget con la lettera F sceglie un nuovo seme e genera la mappa relativa.

Il pulsante *Colors* dà accesso all'omonimo pannello di controllo, per mezzo del quale si può passare da un paesaggio desertico ad uno alpino, o addirittura lunare: basta scegliere opportunamente i colori della palette.

Selezionando i pulsanti *Stars* e *Waves* si aggiungono alla scena rispettivamente stelle ed onde (se è presente dell'acqua), mentre *LockPalette* impedisce future modifiche alla definizione dei colori.

Le caselle DX, DY, DZ e DR contengono la distanza (sui tre assi e reale) tra telecamera e punto di vista, mentre *Bank, Head e Pitch* sono le inclinazioni della telecamera rispetto agli assi.

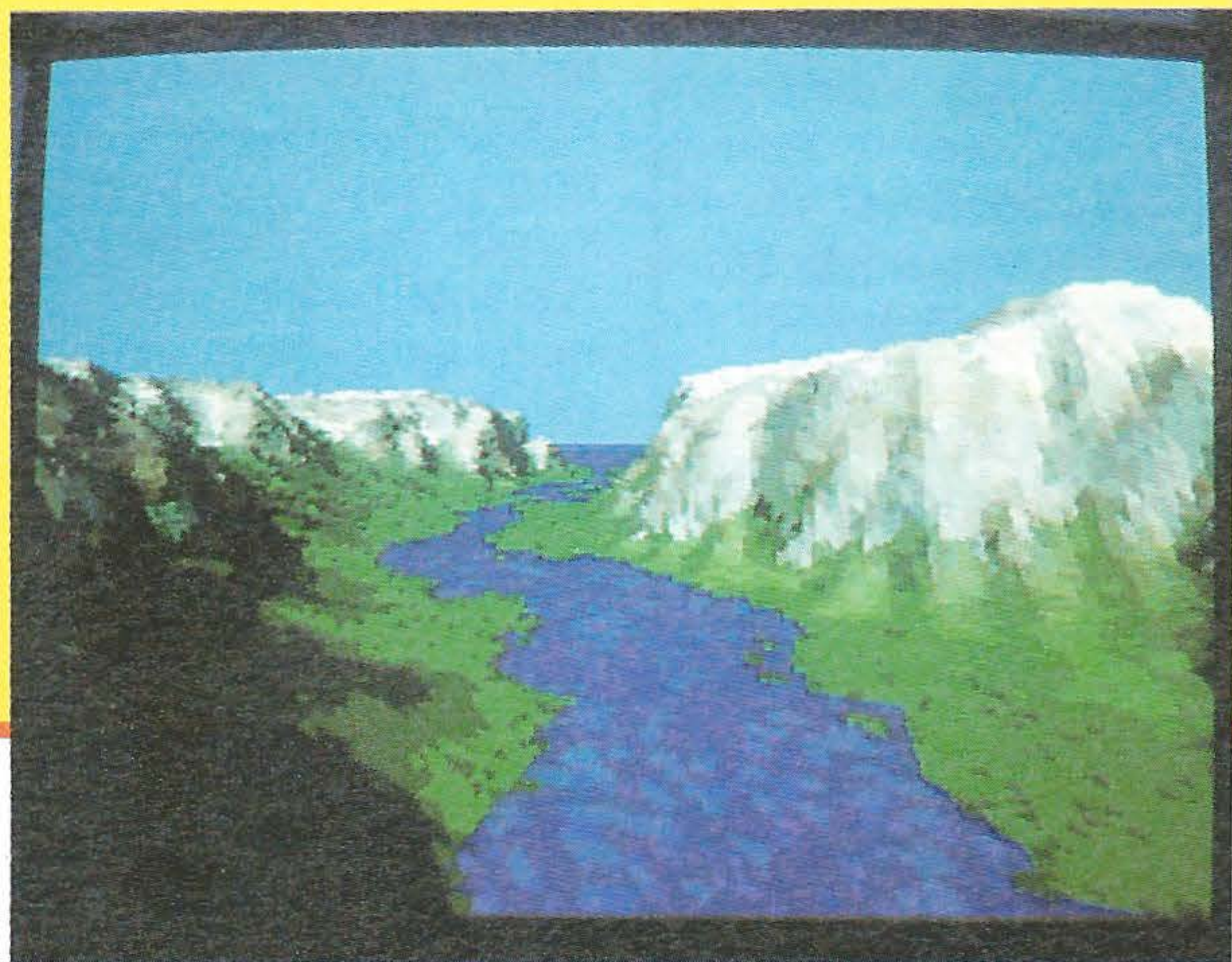
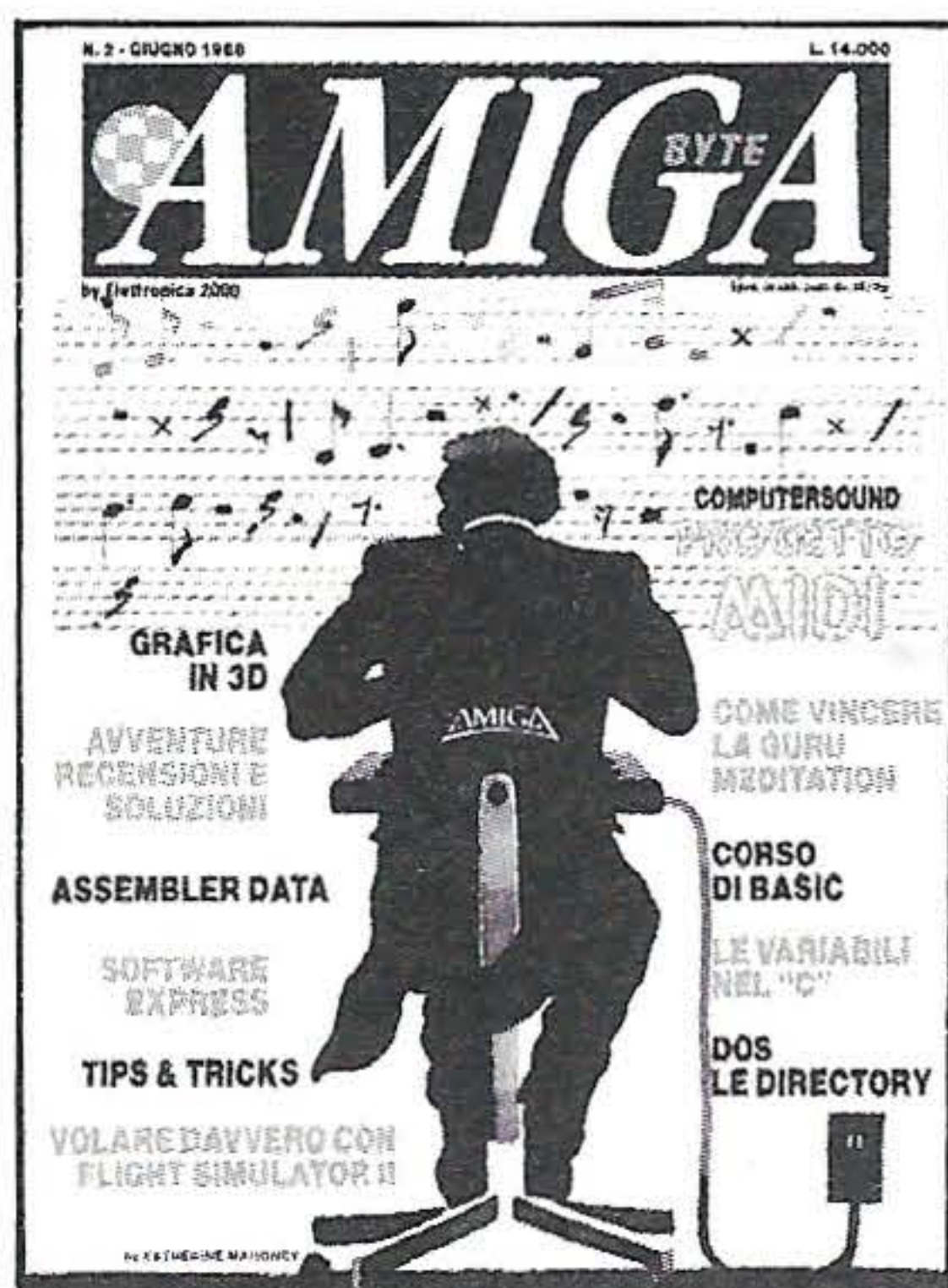
I rimanenti parametri influenzano più nel dettaglio la generazione dell'immagine: *Dithering* aggiunge un fattore casuale all'algoritmo di attribuzione dei colori così che, per esempio, il limite delle nevi perenni non sia una linea continua ma frastagliata; anche *Rough* introduce un fattore casuale, schiarendo o scurendo ogni poligono per dare un'impressione di asperità del terreno.

***Frdim* e *Frdiv*, per finire, sono la dimensione ed il divisore utilizzati nell'algoritmo di interpolazione del programma: è importante che *Frdiv* non sia maggiore della precisione che avete scelto.**

I paesaggi generati da «Vista» sono molto suggestivi ed in genere rispondono in pieno ai desideri dell'utente (al contrario di quanto avviene con altri programmi dello stesso tipo): l'unico neo è rappresentato dalla scarsa definizione del terreno in primo piano, che appare distintamente (anche alla massima precisione possibile) formato da triangoli.

Il risultato del nostro lavoro può essere salvato in formato *IFF* e *Turbo Silver (Imagine)*, e più immagini possono essere generate in sequenza grazie alla capacità di lavorare in modo batch.

Ricordiamo infine che di recente è stata pubblicata una versione potenziata, per Amiga dotati di 68020 o 30, denominata «Vista Pro» che, tra le altre cose, elimina il problema dei triangoli ricorrendo alla tecnica di Gourad.

Anche il comando River fornisce risultati apprezzabili.

si riferirà al primo parametro con la **variabile a1**, e così via fino ad **a<n>**. La memoria per i valori dei parametri è allocata dinamicamente ogni volta che le sotto-funzioni sono chiamate: ciò rende possibile la scrittura di funzioni ricorsive.

Le primitive utilizzabili per la scrittura della propria formula, visualizzabili con il comando **Global / Functions**, sono mostrate in figura, e comprendono tutti i parametri ambiente di «MathVision» (Xmin, Xmax, la larghezza dello schermo, il colore attualmente selezionato, le coordinate in numeri reali o in pixel del punto in esame...) oltre che, naturalmente, tutte le funzioni che un matematico potrebbe desiderare (trigonometriche, iperboliche, booleane; oltre che numerosi tipi di arrotondamenti, ed altro ancora). Per esempio, **Dist(x1,y1,x2,y2)** restituisce la distanza tra due punti sul piano cartesiano, mentre **Radius(x,y)** fornisce la distanza di un punto dall'origine.

Un discorso a parte meritano alcune funzioni che non restituiscono un valore, ma hanno un effetto immediato. Ad esempio, la funzione **Color(n)** fa sì che il punto in esame venga tracciato con il colore <n>: è utile per inventare noi stessi algoritmi di colorazione differenti da quello di default, che dopo un po' rischia di annoiare. Per utilizzare questa funzione basta sommare ad essa (in qualsiasi posizione) il **parametro Null(Color(n))**. La **funzione Null**, come dice il nome, restituisce sempre il valore zero: l'aspetto interessante è che i suoi parametri vengono in ogni caso valutati. Se, come in questo caso, c'è tra di essi un'istruzione, questa viene eseguita come «effetto collaterale». Naturalmente, <n> non deve necessariamente essere un numero: può consistere in un'espressione anche complessa.

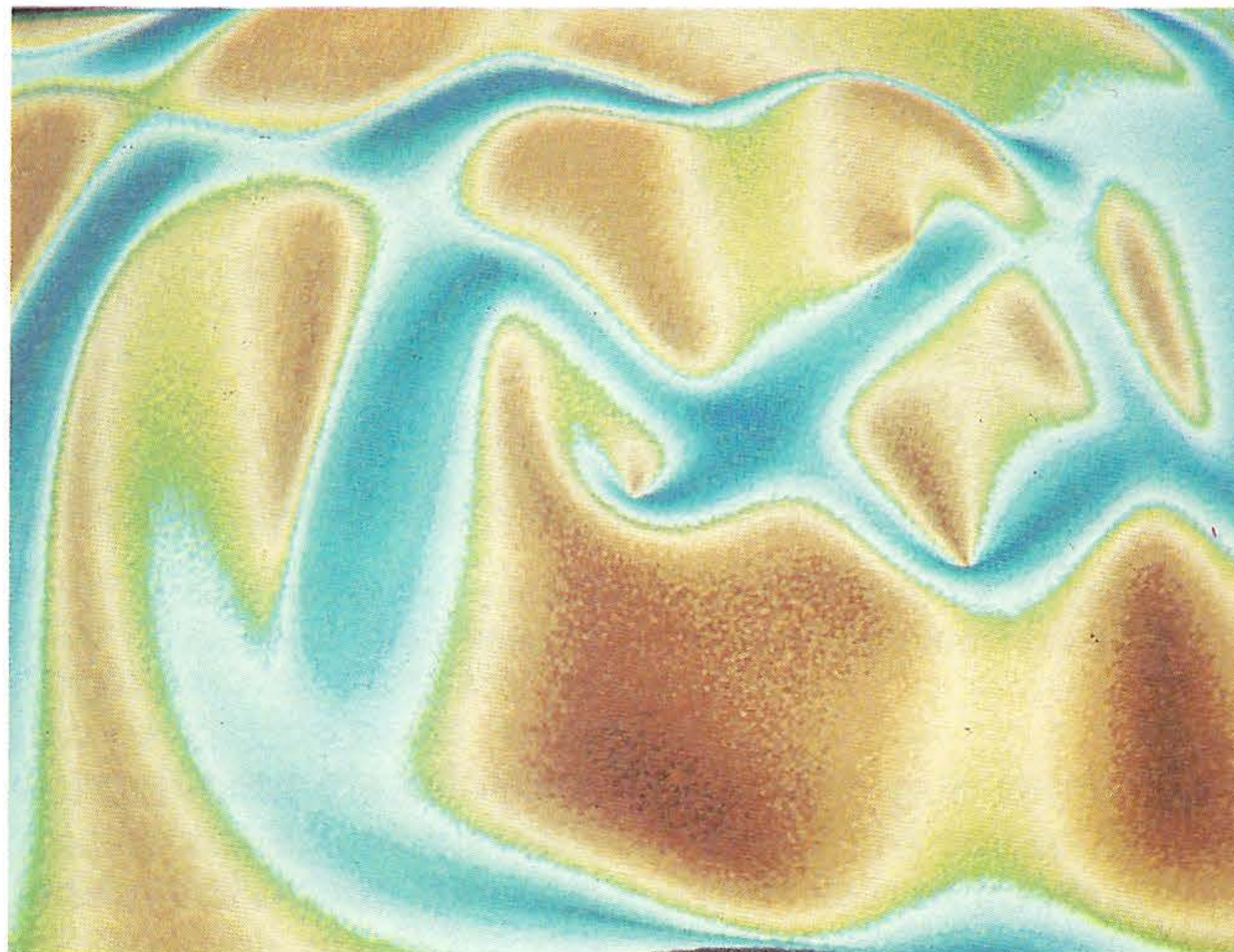



Fig. 3. La stessa funzione della figura precedente, visualizzata con il valido aiuto dell'hook *Shade.*

Fanno parte del gruppo delle funzioni-istruzioni, oltre a **Color**, anche:

Altit(h): setta direttamente la quota del grafico a **h**

Line(x1,y1,x2,y2,n): traccia una linea tra due punti, nel colore desiderato

Draw(x2,y2,n): traccia una linea fino al punto indicato

Wpix(x,y,n): modifica lo stato del pixel di coordinate x,y

Wreal(x,y,n): modifica lo stato del pixel corrispondente alle coordinate *reali* x,y

Wreals(x,y,n): come sopra, ma tiene conto dei parametri Clipping e Scaling.

L'utilità delle ultime quattro istruzioni risulterà

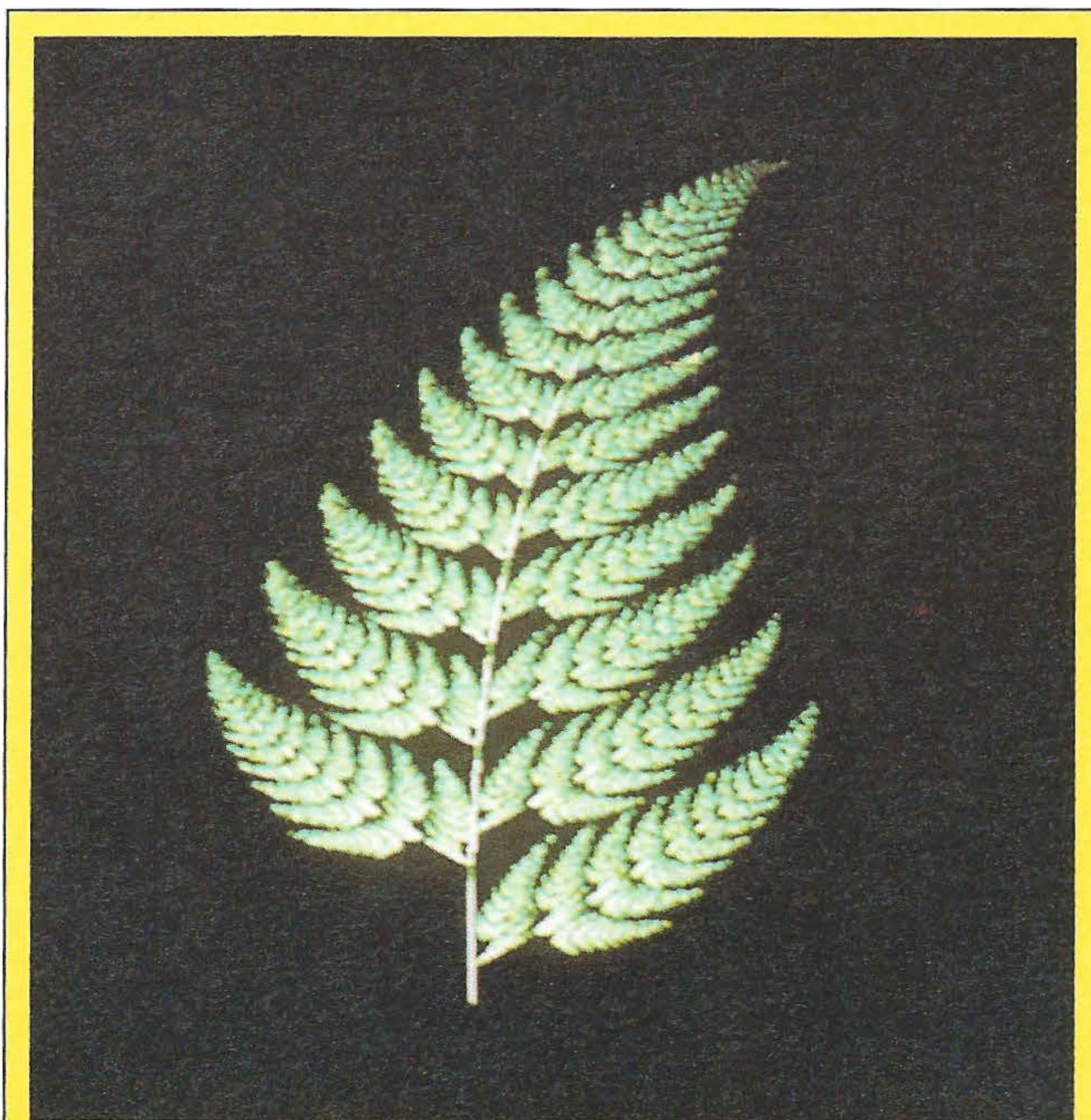



Fig. 4. Un'immagine frattale tra le più comuni: la felce. Questa volta l'hook utilizzato è *Parametric.*

evidente più avanti, quando faremo la conoscenza del **Parametric Plot**.

Un attimo di pausa per una riflessione: «MathVision», solo per quanto visto finora, è sicuramente il programma più potente in commercio per lo studio di funzioni ad una e due variabili. Il bello è che non abbiamo ancora trattato i suoi punti di forza!

Il primo di essi consiste nel meccanismo degli **Hooks**. Questo termine (letteralmente significa «ami») sta ad indicare alcuni programmi esterni che, lanciati in background quando «MathVision» è già attivo, ne aumentano le possibilità aggiungendo nuove funzioni o nuovi comandi, che compaiono nel **menu Hooks**. Del pacchetto fanno parte ben quattordici hooks, nella doppia versione **.ieee** e **.ffp**: ovviamente, occorre utilizzare quelli corrispondenti alla versione del programma che si sta utilizzando.

Cominciamo l'esame degli Hooks dal più spettacolare: stiamo parlando di **Shade**, che si trova nella **directory Plotmethods**. Dopo averlo lanciato, avremo a disposizione un nuovo tipo di grafico; i suoi parametri sono gli stessi fissati per il grafico in prospettiva, ma il risultato che si ottiene con il comando Shade dell'ultimo menu è decisamente più attraente.

Il secondo Hook che prendiamo in considerazione è detto **Parametric**, utile in congiunzione alle funzioni-istruzioni, delle quali si è parlato in precedenza. Esso fornisce un nuovo tipo di grafico (detto appunto **Parametric**) selezionando il quale nessun punto sarà tracciato senza un'apposita istruzione: il programma si limiterà a calcolare infinite volte il valore di **val** (e quindi ad eseguire tutte le istruzioni in esso contenute), finché tale valore non sarà nullo. Questo Hook aggiunge anche una funzione, **count**, che restituisce il numero di volte che la funzione attuale è stata chiamata (utile non solo per il parametric plot, ma anche in caso di funzioni ricorsive).

Focus e **Focusdots** disegnano più grafici di tipo **contour**, decrementando ogni volta il **sample delta** fino al minimo, dando l'impressione di una «messa a fuoco» progressiva dell'immagine.

DA MANDELBROT A J.S. BACH

Anche nella **directory Misc** il materiale interessante non manca: innanzitutto **Mandelbrot**, che aggiunge quattro nuove funzioni per calcolare se una coppia di valori faccia parte o meno degli insiemi di Mandelbrot e di Julia (con numero di iterazioni definito dall'utente), o per determinare il numero di iterazioni oltre il quale la successione associata ad uno di tali insiemi diviene divergente.

È inoltre disponibile un nuovo plot mode, detto **Divide and Conquer**, che dà origine ad un grafico simile al Contour, ma che rende assai meno monotona l'attesa. Esso procede dividendo ricorsivamente lo schermo in due parti finché tali parti non risultano grandi due pixel.

Poiché il calcolo delle funzioni di questo Hook è molto oneroso in termini di tempo, per i sistemi con coprocessore matematico è disponibile la versione **Mandelbrot.68881**.

Il successivo Hook è detto **Sound**, ed aggiunge per l'appunto nuovi comandi per l'emissione di suoni di varia frequenza: **Sound**, **Note**, **Volume** e **Delay**. Purtroppo non è possibile modificare la forma d'onda del suono, ma ci auguriamo che qualcuno realizzi presto un Hook che lo permetta: sarebbe molto bello poter suonare una funzione! **Axes** aggiunge l'omonimo comando, che traccia due assi sullo schermo: è utile soltanto per le funzioni ad una variabile tracciate con i comandi del **menu Simple**. **ProcessPixels** effettua alcune trasformazioni su cia-

Fig. 5. Una bella immagine dimostrativa: figure di interferenza di tre onde sinusoidali concentriche, ancora con *Shade*.

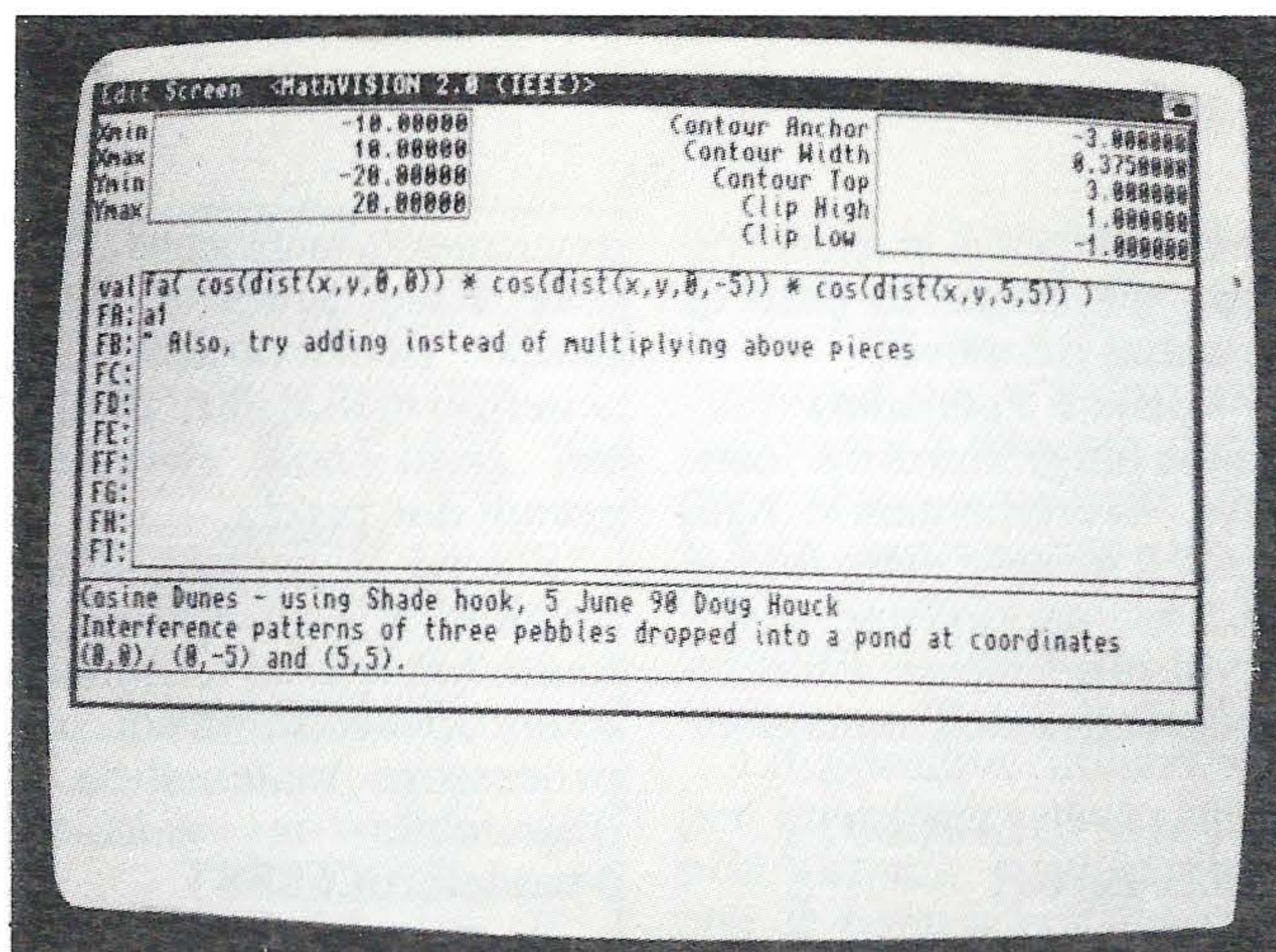
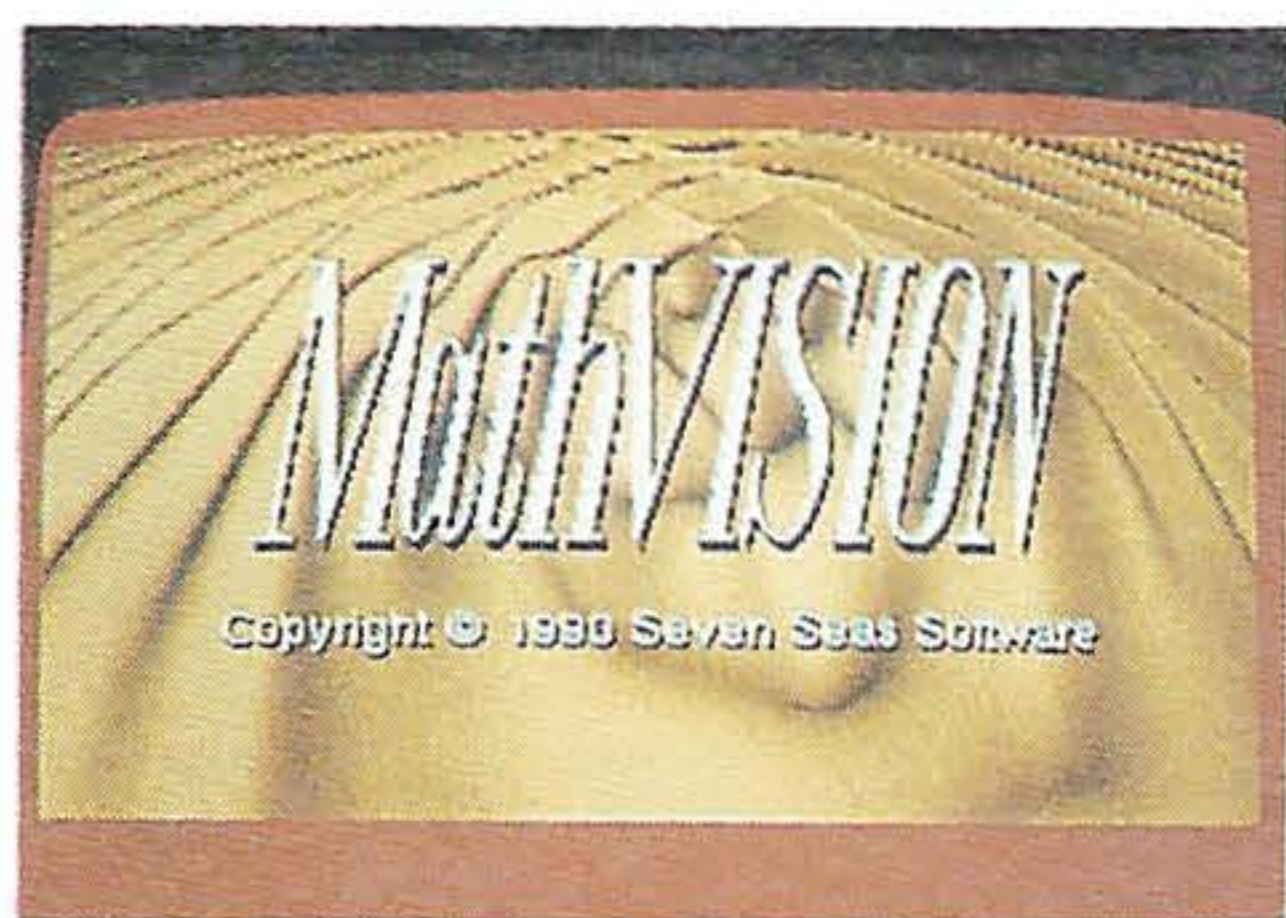



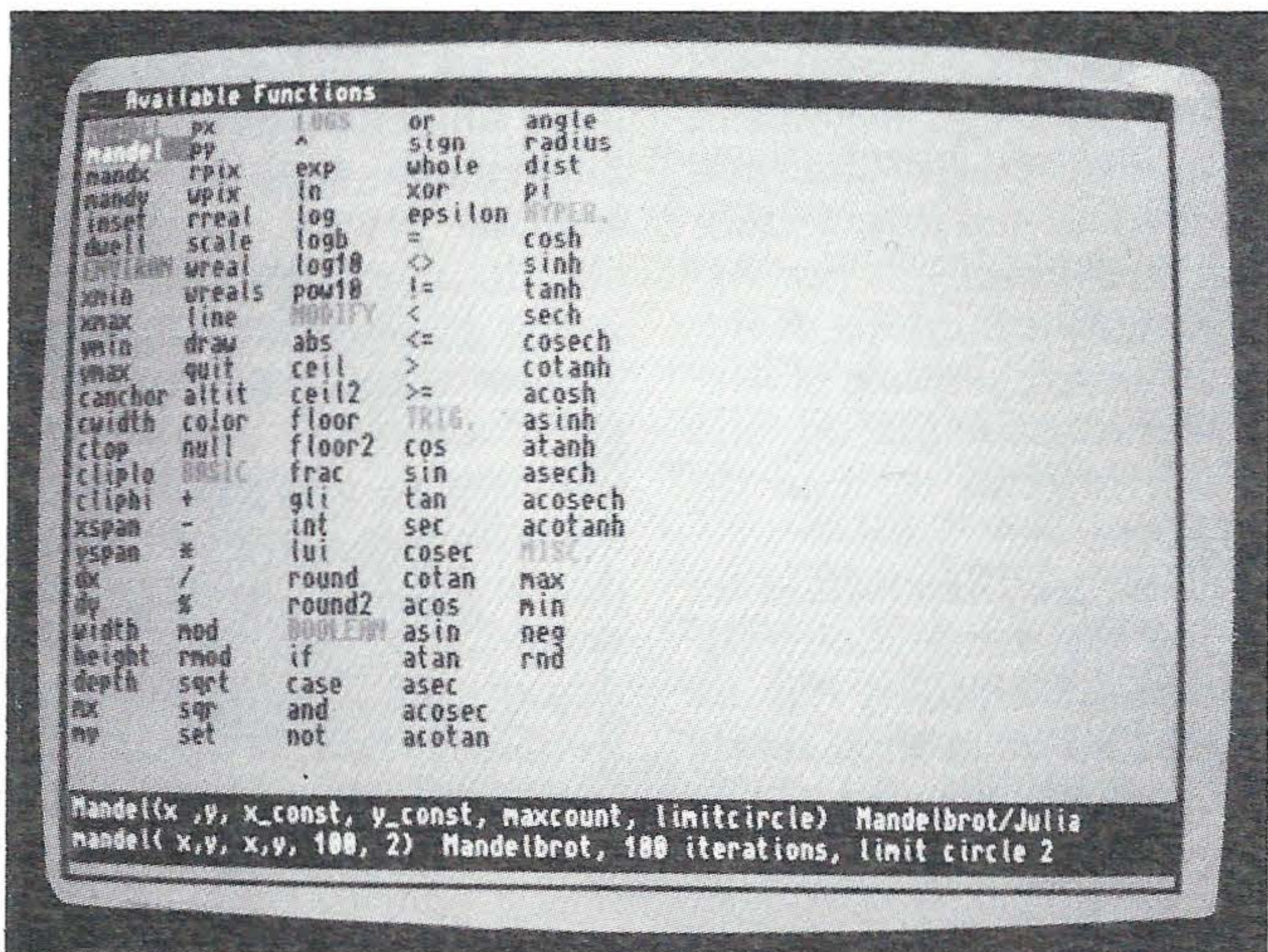
Fig. 6. La funzione che dà origine alla bella figura della pagina precedente. Su di un Amiga 3000 25 Mhz sono necessari circa 15 minuti.

scun punto dello schermo, basandosi sul colore di quelli adiacenti secondo schemi predeterminati, oppure definiti dall'utente: ogni pixel del circondario può avere un peso diffe-

rente nel calcolo del nuovo colore di un punto. Come è facile osservare, si tratta di una funzione che ha ben poco di matematico, e serve semplicemente per ritoccare immagini già pronte. Può essere utilizzata anche su immagini IFF qualsiasi, caricate con l'apposita opzione del primo menu.

Nella directory **Colormapping** troviamo tre Hooks che permettono, tramite pannelli di controllo esterni a «MathVision», di modificare gli algoritmi di selezione colori. In particolare, due di essi lavorano in modo HAM, e soltanto per grafici di tipo Contour.

Nella **directory Storage**, infine, **Zbuffer** costruisce una matrice bidimensionale nella quale, per mezzo di apposite funzioni, è possibile inserire dei dati. L'ultimo Hooks, **DataIO**, somiglia molto al precedente ma ha in più la possibilità di memorizzare su disco la matrice attuale, in formato ASCII o IFF.

Il manuale comprende alcune pagine di informazioni, che permettono di costruirsi i propri Hooks.

Fig. 7. L'elenco delle funzioni elementari disponibili. Le prime sono quelle aggiunte dall'hook *Mandelbrot*.

PORTE APERTE

E per chi non conosce il C? Tranquilli, gli Hooks non sono l'unica possibilità per interagire con «Math Vision»: come tutti i programmi più recenti, infatti, esso è dotato di una porta **ARexx**, per mezzo della quale può essere pilotato come e meglio che via mouse.

Ogni utente, purché possieda una minima conoscenza in materia e l'interprete di tale linguaggio (incluso tra l'altro nel WorkBench 2.0) può realizzare programmi personalizzati per qualsiasi sua esigenza: poiché sono previsti sia l'input che l'output di tutti i parametri, si può riprogettare completamente il pannello di controllo del programma, costruendone uno su misura per le proprie esigenze.

COLORI, BIORITMI E ANIMAZIONI

Gli esempi forniti sono numerosi e ben commentati, quindi facilmente modificabili: si va dal programma che sfuma i colori a quello che traccia i bioritmi, a quelli che realizzano animazioni.

Del pacchetto fa parte un'utility di pubblico dominio, chiamata **RecordAnim**, che consente (per mezzo di semplici comandi ARexx) di creare file in standard **IFF Anim** che potranno essere letti anche da altri programmi.

Per qualsiasi immagine, il comando **Print Text** del primo menu fornirà (su stampante) l'elenco dei comandi ARexx necessari per realizzarla.

La porta ARexx fa di «MathVision» l'ideale candidato per tutorial interattivi sulla matematica, possibilmente in congiunzione con un programma per il Multimedia Authoring come «**AmigaVision**» o «**CanDo!**»: ci auguriamo che qualcuno si dia presto da fare in questo senso.

MATEMATICA E ARTE

Il terzo dischetto contiene una galleria di esempi ampiamente commentati, alcuni dei quali davvero stupefacenti, realizzati con il solo programma o con l'aiuto di vari Hooks. Tra di essi, i lettori più assidui di AmigaByte riconosceranno una variazione (a colori!) sul tema della «felce», figura geometrica frattale trattata sul numero 20 di marzo 1990.

In conclusione, «Math Vision» non è un programma le cui potenzialità possano essere individuate immediatamente: l'abbondanza delle sue funzioni e soprattutto la sua struttura aperta ne fanno un software quasi universale. A scanso di equivoci, tuttavia, sarà meglio vedere subito cosa *non* può fare. Non si tratta di un software di matematica in senso stretto: non vi aspettate che possa risolvere un sistema di equazioni, o calcolare un integrale, o gli asintoti di una funzione. Purtroppo, per Amiga non sono ancora diffusi programmi di questo tipo, che pure godono di un certo seguito su Macintosh e Ms-Dos.

Se il vostro obiettivo è lo studio dell'insieme di Mandelbrot così com'è, senza elaborazioni grafiche, sappiate che «MathVision» è molto più lento di altri programmi, anche di pubblico dominio, a ciò dedicati (per esempio «**Mandelbrot Explorer**» o «**ImandelVroom2.0**»). Se invece desiderate usare l'insieme come base per successive elaborazioni matematiche, non avete altra scelta: ma credeteci, sarà un obbligo piacevolissimo!

Un ritorno alla grande: The Director 2.0

Il popolare linguaggio per la creazione di animazioni e di presentazioni grafiche e sonore torna al meglio di sé sugli schermi di Amiga in una nuova veste potenziata.

di FRANCESCO ANNONI

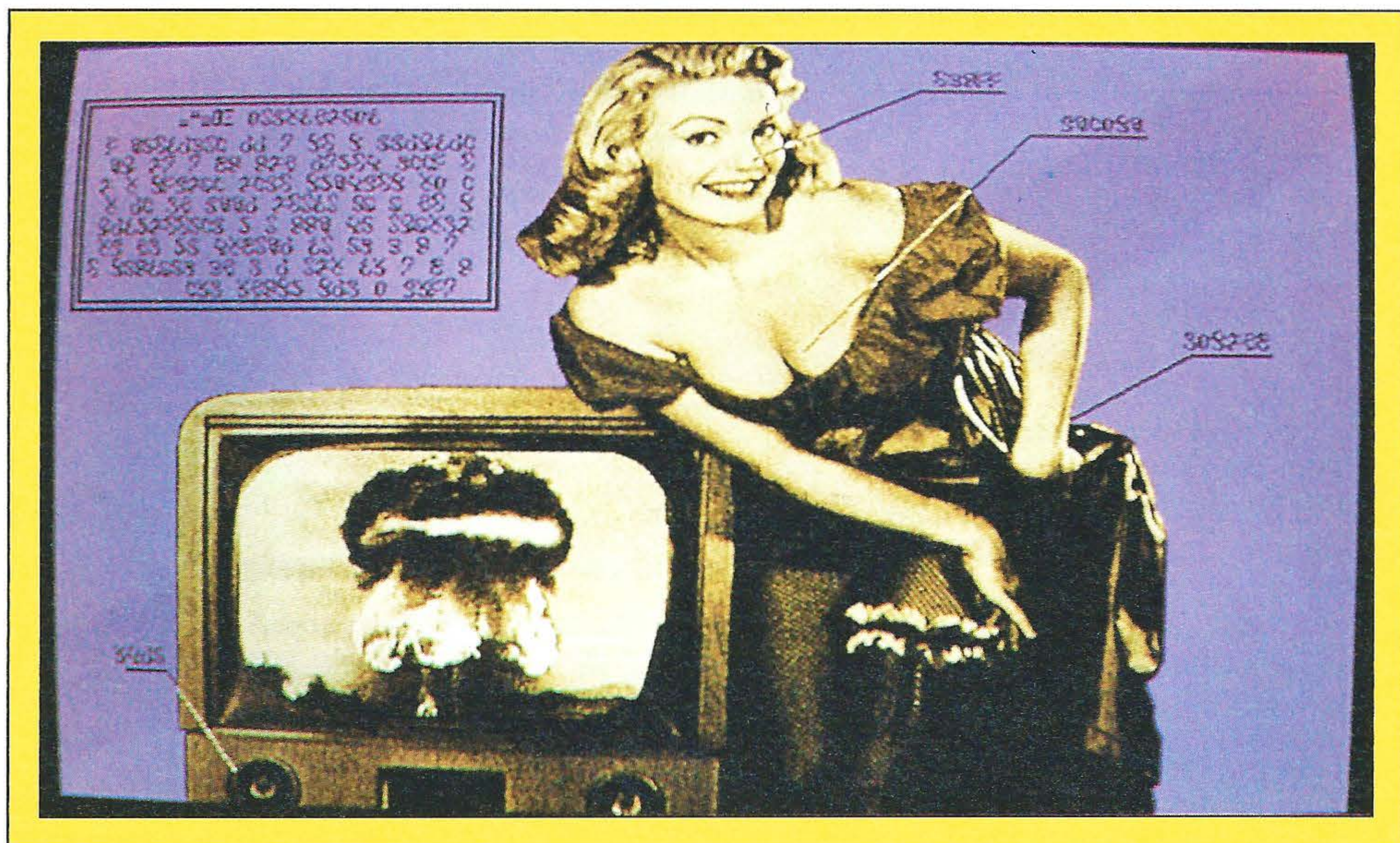
Alcuni programmi sono come i divi di Hollywood: nonostante il passare degli anni rimangono miracolosamente giovani. «**The Director**» è sicuramente uno di questi: vuoi per l'abilità dei programmatori, vuoi per la mancanza di veri e propri concorrenti, ha rappresentato a lungo uno strumento utilissimo per i computergrafici.

A quasi quattro anni dalla nascita, il programma è stato ampliato ed aggiornato, assumendo una veste sicuramente più professionale della precedente.

Per chi ancora non lo conoscesse, ne descriveremo brevemente la filosofia di lavoro: «The Director» è essenzialmente un linguaggio di programmazione, concepito per svolgere tutti quei compiti (e sono tanti!) che non possono essere portati a termine con un qualsiasi programma preconfezionato.

Non ha requisiti minimi (il che significa che può essere utilizzato perfino su di un 500 inespanso), ma lavora in perfetto accordo con qualsiasi cosa in più voi possediate (espansioni, hard disk, schede acceleratrici): in particolare gira benissimo su Amiga 3000 sotto Kickstart 2.0, del quale permette anche di sfruttare alcune delle nuove caratteristiche.

UNO STRUMENTO IDEALE

È orientato principalmente alla grafica ed al sonoro, ma con un occhio di riguardo per tutte le altre capacità di Amiga, il che ne fa uno strumento ideale per costruire da sé presentazioni, dimostrazioni, corsi, e chissà che altro. Dopo

avere scritto il testo del programma con un qualsiasi editor (ma la nuova versione comprende un potente editor integrato), il programma «**Director**» si occupa di verificarne la correttezza e di tradurla in un formato denominato **.film**. Tale formato, illeggibile per gli esseri umani, viene letto ed interpretato dal programma «**Projector**», che si occupa dell'effettiva esecuzione dei nostri ordini ed è, in realtà, l'interprete propriamente detto; ma, essendo esso inservibile senza l'accoppiamento a «Director», può essere liberamente duplicato e distribuito insieme alle nostre creazioni.

Detto ciò, osserviamo subito che il nuovo «The Director 2.0» traduce senza difficoltà anche listati scritti per il suo predecessore, mentre non è così per «Projector». In altre parole, un file preesistente in formato **.film** dovrà essere eseguito con la versione corrispondente di «Projector».

APRIAMO I DISCHETTI

Dunque, apriamo i dischetti. Niente paura, si tratta di apertura mediante doppio click, e non fisica: nei due dischi del pacchetto troviamo, oltre che l'editor dedicato «**DEdit**» ed i suddetti programmi, ribattezzati «**D2**» e «**Proj2**», la directory **Modules**. Quest'ultima contiene parti dell'interprete dedicate a compiti specifici, che vengono caricate solamente se necessario, in seguito ad una precisa richiesta del programmatore (vedremo più avanti due esempi di utilizzo di Modules): questo approccio risulterà particolarmente gradito a chi non possiede un hard disk, dato che la durata del caricamento di «Proj2» ne viene sensibilmente ridotta.

La directory **Libraries**, invece, contiene librerie di funzioni *da includere* nei propri programmi: ciò significa che queste funzioni sono scritte nel linguaggio di «The Director», e che possono essere esaminate ed eventualmente modificate (fatelo soltanto quando sarete abbastanza esperti nell'uso del programma!).

La directory **Tools** comprende alcune utility esterne, che servono a velocizzare alcuni aspetti della programmazione. Volete trarre vantaggio dalle avanzate funzioni di elaborazione di poligoni che «The Director» vi mette a disposizione, ma non avete nessuna voglia di ricavare le coordinate di ogni vertice? La «**PolyUtil**» (Polygon Utility) è quello che fa per voi.

Prima di includere un file musicale in formato **.smus** nel vostro programma volete sapere di quali strumenti avrà bisogno? Vi viene in aiuto l'impronunciabile «**Exsmus**»; e così via. Un apposito file (di testo, quindi modificabile) permette di aggiungere ai menu dell'editor nuove utility per richiamarle direttamente da esso.

Incidentalmente, notiamo che il comando associato alla PolyUtil ha la forma **Proj2 Tools/Polyutil.film**: se ne deduce che anche le utility esterne sono state realizzate con «**Director**» stesso. Questo sì che è fare le cose in famiglia!

Le directory rimanenti si chiamano **Scripts**, **Sounds**, **Pictures**, **Music**, **Instruments**: la prima contiene i programmi veri e propri, per le altre il nome dovrebbe essere sufficientemente esplicativo.

I programmi dimostrativi sono ben quarantuno, e coprono esaurientemente tutte le possibilità del linguaggio Director: dall'uso di moduli e librerie, all'esecuzione di musica o di animazioni (in formato Anim o mediante page-flipping), al caricamento di font esterni.

VISITA GUIDATA

Entriamo nell'editor, con un doppio click sulla sua icona (a differenza della versione precedente, la 2.0 è guidata completamente con il mouse): compare lo schermo in figura 1.

Chiunque abbia usato una volta nella vita un editor di testi (come «**MicroEmacs**» o «**CygnusEd**») non avrà nessuna difficoltà a comprendere l'utilizzo del-

```
Director_esempio
    LOAD 2,"animagic:unfoldx4.anim"
                    : rem carica il primo fotogramma
    copy 2,3        : rem lo ricopia in un altro buffer
    LOADANIM 1,"animagic:unfoldx4.anim"
                    : rem carica l'animazione
    display 2       : rem mostra il primo buffer,
    b=3             : rem ma lavora sul secondo
/loop:
  ANIM 1,b,done     : rem costruisce il prossimo fotogramma
  if done then SKIPANIM 1
                    : rem era l'ultimo? Ricomincia da capo
  DISPLAY b         : rem lo visualizza
  b=5-b             : rem lavora sull'altro
                    :     che ora e' nascosto
  GOTO loop
```

Fig. 1. Poche istruzioni sono sufficienti per realizzare un flessibile player di animazioni.

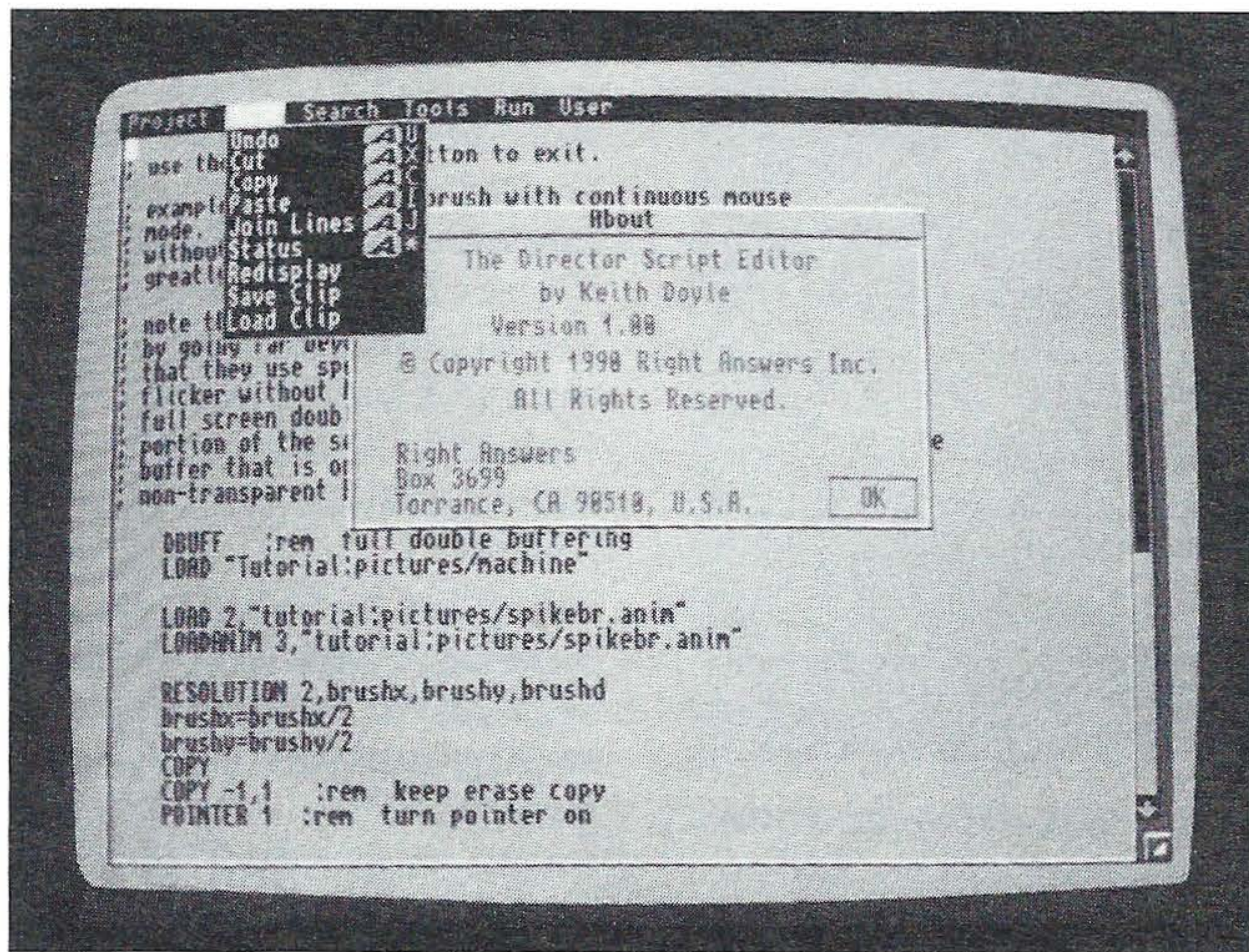



Fig. 2. L'Editor dedicato DEdit non è potentissimo, ma è indubbiamente molto comodo.

le funzioni dei primi tre menu: **Project**, **Edit** e **Search**. È perciò sugli altri tre che la nostra attenzione si concentrerà.

Sotto la voce **Tools** troviamo alcune opzioni che rendono meno monotono il nostro lavoro di programmatori. Innanzitutto **Library**, che aggiunge un'opportuna istruzione **Include** per rendere disponibili al programma le istruzioni aggiuntive contenute nelle librerie; **View IFF**, per dare un'occhiata ad un'immagine senza bisogno di creare uno script che la carichi; **Coordinates** (attivo solamente dopo l'uso di **View IFF**), che fornisce le coordinate di schermo di un punto indicato con il mouse.

Select File è un comando per pigri e smemorati: se non avete voglia di inserire un nome di file completo di path (o se non lo ricordate), potete selezionarlo tramite un file requester e verrà inserito nel programma.

Il comando **Use Segment**, infine, include le istruzioni contenute nel file RAM:Segment; tale file è inizialmente vuoto, ma può venire creato e/o modificato da utility esterne.

Il quinto menu comprende solamente il comando **Run**, che sottopone il nostro script a «Director» e successivamente (se tutto è andato bene) a «Projector», il tutto in modo completamente automatico. È dunque possibile passare semplicemente e rapidamente dalla fase di scrittura a quella di test del nostro programma.

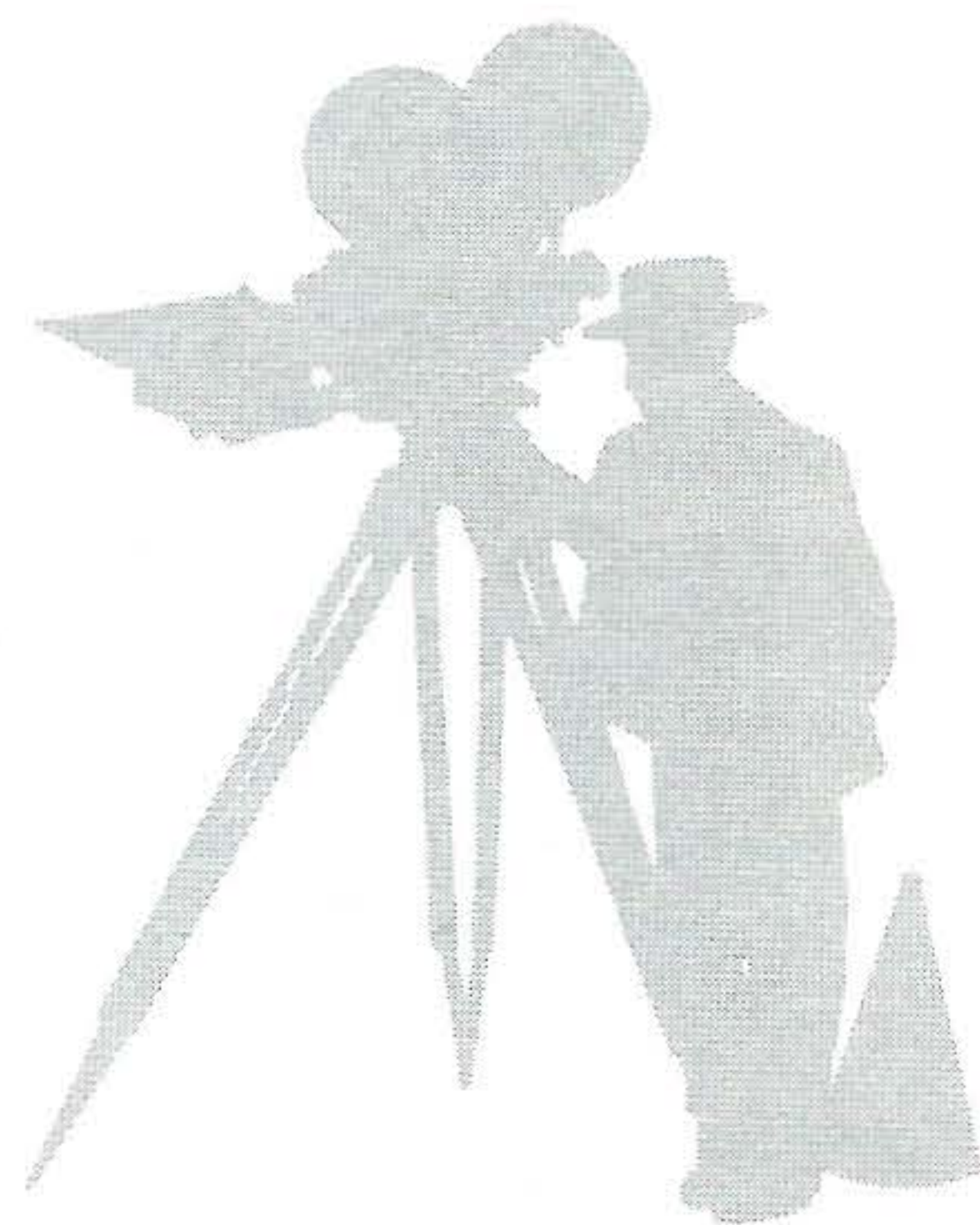
L'ultimo menu è denominato **User**, e non a caso: i suoi contenuti sono decisi dall'utente, che vi può inserire comandi per richiamare qualsiasi utility richiamabile da CLI. Nella configurazione di base sono presenti la **Blit Utility**, per la scrittura automatica di parti di programma riguardanti operazioni di sovrapposizione e ritaglio di immagini sullo schermo, e la **Button Utility**, per la creazione di pulsanti e di primitive icone da azionare con il mouse. Ad esse si accompagnano le già viste **Polygon Utility** e **ExSmus**.

Il file **Tools/DEdit.User.Menu**, mediante il quale ciascuno può configurare l'editor a seconda delle sue esigenze, ha questa struttura: nella prima riga, il numero di voci che tale menu deve contenere; le righe seguenti contengono il testo che dovrà apparire nel menu, ed il comando Shell ad esso associato, separati da una virgola, per ciascuna delle voci. Per aggiungere un comando atto ad aprire una nuova Shell, ad esempio, occorrerà incrementare il numero sulla prima riga (da 4 a 5), ed aggiungere la riga **New Shell, Newshell** in fondo.

Fatta la necessaria conoscenza dell'ambiente di lavoro, esaminiamo più da vicino le possibilità che il linguaggio Director ci offre.

ISTRUZIONI PER L'USO

Le istruzioni base per la grafica sono:

Load [<n>],<nomefile> carica un'immagine IFF, eventualmente attribuendole il numero n;
New <width>,<height>, <depth> crea una nuova pagina grafica, delle dimensioni specificate;
Display [<n>] visualizza l'immagine appena caricata (la numero n, se tale parametro è presente);
Copy <n>,<m> crea l'immagine #m, copia esatta della #n.
Loadanim <n>,<nomefile> (carica un'animazione ANIM attribuendole il numero n);
Anim <n>,[<m>],done (calcola il prossimo fotogramma dell'animazione #n, lo copia nel buffer per immagini #m, e setta la variabile **done** a 1 se il fotogramma era l'ultimo;
RevAnim <n>,[<m>], done (come sopra ma calcola il fotogramma precedente);
Skipanim <n> passa dall'ultimo al primo fotogramma dell'animazione #n. Se si sta usando **RevAnim** per far andare l'animazione all'indietro, la sintassi si complica leggermente e diviene **Skipanim <n>,–2.**

Con le medesime istruzioni si gestiscono gli Anim Brush, chè questi ultimi altro non sono che animazioni di formato ridotto.

Alla luce di quanto visto finora, un semplice player per animazioni si può realizzare con i comandi di figura 2. In caso di animazioni particolarmente lunghe, è possibile caricarle in background: il comando è

Fig. 3. Panoramica delle decine di script dimostrativi presenti sul disco Tutorial.

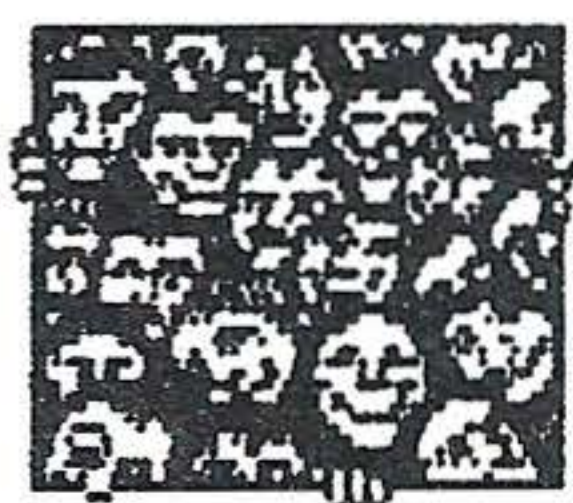
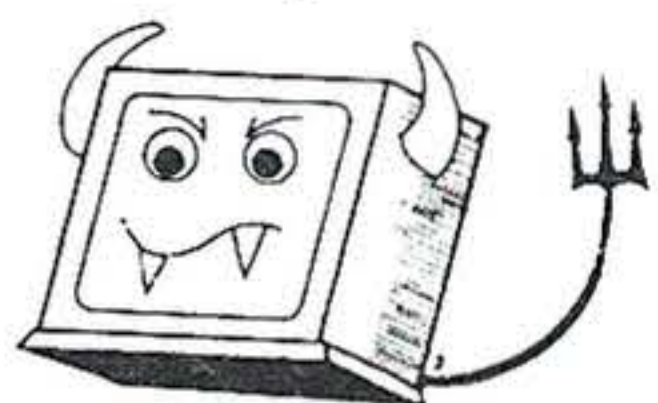



Fig. 4. Una veduta parziale dei contenuti delle numerose directory del pacchetto.

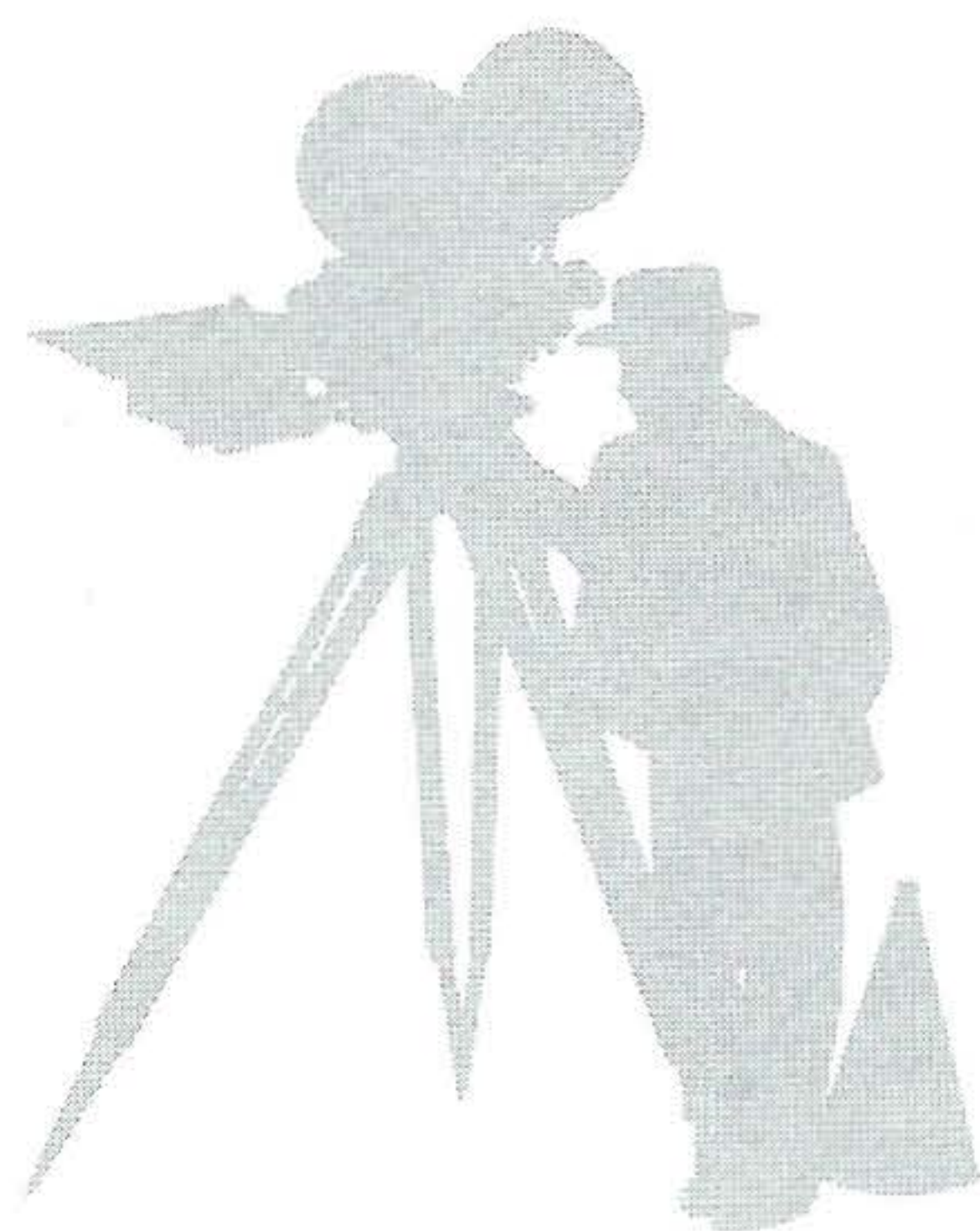
Bloadanim <n>, "nomefile". Per verificare se in un dato momento un fotogramma è già stato caricato utilizziamo **Bcheck v,1,<frame>**: la variabile v sarà posta a 1 in caso di esito positivo.

ALTRI COMANDI GRAFICI

Il comando **Pen <n>,<c>** attribuisce il colore numero <c> alla penna specificata dal parametro <n>: la penna 1 è quella normalmente usata da tutte le istruzioni grafiche, la 0 è per gli sfondi, la 2 per i contorni.

Il comando **Resolution <n>, width, height, depth** inserisce nelle tre variabili le dimensioni dell'immagine contenuta nel buffer #n; **Move <x>,<y>** imposta le coordinate per la successiva istruzione grafica; **Draw <x,y>** traccia una linea fino al punto specificato, mentre **Text «Amigabyte»** scriverà questa parola tanto musicale nella posizione attuale. **Loadfont <n>, <h>,"nome.font"** carica un font carattere, assegnandogli il numero <n>; **Setfont <n>** fa sì che il font caricato con il numero <n> venga utilizzato per le prossime istruzioni **Text**. Per scrivere un messaggio senza utilizzare lo schermo del programma si può usare l'istruzione **Note**: l'output verrà diritto ad una finestra CLI (se esiste), o ad un requester aperto per l'occasione.

L'istruzione **Circle <x>,<y>,<r>, <fill>** traccia una circonferenza nel colore della penna n. 1 (o un cerchio, se il parametro <fill> vale 1: in questo caso, il colore della circonferenza è dato dalla penna n. 2); dei rettangoli si occupa **Rect <x>,<y>,<x2>, <y 2>**.

Normalmente, tutte le istruzioni grafiche hanno effetto sulla pagina attualmente visualizzata. Se lo si desidera, per mezzo del comando **Outbuff <n>** si può agire si di una qualsiasi delle altre (purché esistente).

L'istruzione **Module "Director:Modules/Sound"** aggiunge al linguaggio un'intera famiglia di istruzioni: vediamone le principali.

Sound "Load",<n>,"Nomefile" carica un suono campionato (in formato IFF) assegnandogli il numero <n>;
Sound "Loadfast",<n>, "Nomefile": come sopra, ma il suono viene caricato in FAST RAM e verrà spostato in CHIP solo al momento dell'esecuzione;
Sound "Play",<n>,<rip>, <vol>,<start>, <chan> manda l'effetto sonoro numero <n>, ripetuto <rip> volte, in uscita sul canale <chan>, al volume <vol> (da 0 a 64). Se il parametro <start> viene posto a −1, la riproduzione non ha inizio immediatamente, ma solo con l'apposito comando **Sound "Start"**;
Sound "Stop" [,<n>] zittisce immediatamente il canale audio <n> se specificato, altrimenti agisce su tutti i canali attivi;
Sound status, "Chanstat" [,<n>] assegna alla variabile status (che può avere un qualsiasi altro nome) il valore 1 se il canale <n> è attivo, ed il valore 0 in caso contrario. Se <n> non è specificato, controlla tutti i canali, e pone la variabile a 1 se almeno uno di essi è attivo;
Sound "Slowfade", <vel>,<vol>,<chan> porta lentamente il volume del canale <chan> da quello attuale a <vol> alla velocità indicata da <vol> (0=massima). Si tratta di un processo asincrono: le istruzioni successive vengono quindi eseguite mentre la variazione di volume è ancora in corso.
Sound status,"Slowstat" [,<n>] agisce come la **Chanstat**, ma serve per controllare se c'è qualche fade tuttora in corso.

IL MODULO MUSICALE

Le capacità audio di «The Director» non sono limitate a singoli effetti so-

nori: interi brani musicali (in formato IFF SMUS) possono essere caricati con l'ausilio del **modulo Smus**. Includendo questo modulo nel programma, avremo a disposizione ben 32 istruzioni supplementari, dedicate all'esecuzione di musica: diamo un'occhiata alle principali.

Smus "Load",<n>,"nomefile" carica in RAM un brano, insieme a tutti gli strumenti di cui esso necessita;

Smus "Play",<n> attiva l'esecuzione del brano caricato con il numero <n>;

Smus "ScoreVol",<vol> modifica il volume dell'output sonoro.

Con altre istruzioni di questo modulo possiamo sincronizzare qualsiasi cosa (ad esempio un'animazione) con un particolare strumento: a questo scopo, si usano le istruzioni **Smus buf, "Findbuff", "NomeStrumento"** (che mette nella variabile *buf* il numero associato allo strumento specificato), **Smus "Setsig",buf** (che attiva il controllo di quella traccia audio) e **Smus "Waitsig"**, che attende il suono dello strumento scelto. Ad esempio, sincronizzando un'animazione con la traccia "drumkit" di un brano, potremo essere ragionevolmente sicuri che l'animazione seguirà il ritmo della batteria.

L'istruzione **Getkey A** attende che venga premuto un tasto e pone il codice corrispondente nella variabile A, mentre **Ifkey A** verifica semplicemente se un tasto è premuto in questo momento, senza attendere, comportandosi analogamente alla Inkey$ del BASIC.

Nella versione 2.0 sono state implementate tutte le strutture di controllo che un programmatore possa umanamente desiderare: oltre che la **for-next** e la **if-then** (quest'ultima ha ora sintassi identica a quella BASIC), troviamo la **while-wend** e la **repeat-until**, per eseguire un gruppo di istruzioni rispettivamente fino a quando una data condizione rimarrà vera, o falsa.

Un numero pseudocasuale da 0 a n si indica con **?n**, mentre un'etichetta, o label, è specificata con la sintassi **/Etichetta:** (naturalmente, è richiamabile con **Goto** e con **Gosub**). Uno dei più importanti miglioramenti introdotti dalla versione precedente riguarda le variabili con indice. Gli «aficionados» di «**The Director**» ricorderanno che ne esisteva una sola, monodimensionale, detta @: ora l'istruzione **Dim** (con sintassi simile a quella BASIC) è stata implementata in pieno, e si possono creare array multidimensionali con estrema facilità.

Le stringhe di testo vengono considerate, un po' come in C, array di variabili intere: vanno quindi dimensionate in anticipo, per esempio con **Dim Stringa(100)**. In seguito, la stringa associata a tale array sarà indicata con **Stringa$**.

La descrizione che abbiamo fatto dei principali comandi di «**The Director 2.0**» non pretende certo di costituire una trattazione completa del linguaggio, bensì di dare a chi non lo sapeva un'idea di cosa può fare, e mettere in grado chi lo possedesse già di iniziare a lavorare. Gli innumerevoli script presenti sul disco **Tutorial** costituiscono un validissimo ausilio per l'apprendimento del linguaggio, ma per scatenare tutte le capacità di «The Director 2.0» il manuale è indispensabile: la smisurata potenza del programma costituisce così la miglior forma di protezione contro la pirateria. L'espandibilità futura è garantita dall'impiego delle Library e dei Module. □

Cicli e salti condizionati

Impariamo come impostare in Assembler cicli, confronti e condizioni, con l'ausilio dei flag del registro di stato.

di PIERLUIGI MONTANARI
Quarta puntata

Nella puntata precedente del nostro corso di Assembler abbiamo visto come procedere per visualizzare una riga di testo sullo schermo mediante le funzioni della libreria Dos (dos.library). Questa volta creeremo un programma più complesso, la cui stesura sarà resa molto più veloce grazie all'utilizzo di una nuova serie di istruzioni di fondamentale importanza: quelle che permettono di generare anche in Assembler i cicli FOR ... NEXT del Basic, e strutture complesse come quelle di tipo IF...THEN. Prima di prenderle in esame, vediamo la ragione della loro importanza, riprendendo in considerazione il listato della scorsa puntata, che permetteva di stampare una riga sullo schermo.

UN PASSO INDIETRO

La struttura del programma era, a grandi linee, questa:
1) Apro la Libreria Dos
2) Ottengo l'handle di output (cioè la finestra in cui scrivere i dati)
3) Scrivo la stringa
4) Chiudo la libreria ed esco

Immaginiamo però ora di avere più stringhe da scrivere sullo schermo, una di seguito all'altra: i modi per ottenere questo risultato sono numerosi, il più scomodo dei quali consiste nel ripetere il punto 3 tante volte quante sono le stringhe, cambiando ogni volta il puntatore alla stringa da mostrare.

Il motivo è semplice: ripetere questa sequenza (vedi

```
jsr  output(a6)            ;handle del file
move.l    d0,d1            ;
move.l    #Text,d2         ;indirizzo del testo in d2
moveq     #10,d3           ;lunghezza testo in d3
move.l    DosBase,a6       ;indirizzo dos.library in a6
jsr  write(a6)             ;esegui la funzione write
```

Fig. 1: Questa routine va bene per stampare una riga di testo, ma diventa scomodo ripeterla quando le righe sono numerose.

fig. 1) sette od otto volte di seguito allungherebbe notevolmente il listato, ne appesantirebbe la lettura, e soprattutto occuperebbe molto più spazio del necessario in memoria. Una prima soluzione a questo problema sarebbe quella di isolare una parte del listato, magari contrassegnandola con una label (etichetta) denominata «Print», e limitandosi a fare una serie di chiamate a quella routine variando di volta in volta il puntatore nel registro D2 (assumendo che le linee abbiano tutte la medesima lunghezza). Un esempio è mostrato in figura 2.

```
...
Print:jsr     output(a6)
     move.l   d0,d1             ;metti l' handle in d1
     moveq    #20,d3            ;lunghezza testo in d3
     move.l   DosBase,a6        ;indirizzo dos.library in a6
     jsr  write(a6)             ;esegui la funzione write
     rts
...
Text1:dc.b    "ciao caro amico... "
Text2:dc.b    "come stai??        "
Text3:dc.b    "spero bene !!!     "
...
     move.l   #text1,d2
     jsr      print
     move.l   #text2,d2
     jsr      print
     move.l   #text3,d2
     jsr  print
...
```

Fig. 2: Abbiamo attribuito una label alla routine di stampa per richiamarla variando ogni volta il puntatore alla stringa.

Questa soluzione è certamente da preferirsi alla precedente, anche se non è la più comoda per altre ragioni: innanzitutto perché non è detto che tutte le stringhe da visualizzare abbiamo la stessa lunghezza, ed in secondo luogo perché, dovendo stampare del testo, sarebbe a dir poco stressante dover ogni volta specificare l'indirizzo di una linea di testo, eseguire il salto, etc.

Il metodo risulta invece pratico se le righe sono al massimo tre o quattro, e se hanno all'incirca la stessa lunghezza. L'ideale sarebbe perciò poter fare un ciclo FOR

```
...
Print:jsr      output(a6)
     move.l    d0,d1          ;metti l' handle in d1
     moveq     #21,d3         ;lunghezza testo in d3
     move.l    DosBase,a6     ;indirizzo dos.library in a6
     jsr  write(a6)           ;esegui la funzione write
     rts
...
Texts: dc.b    "Programma fatto da ",10 ;**
       dc.b    "-> Mickey Mouse <-",10
       dc.b    "    Disneyland    ",10
               ...
** N.B.:il numero 10 serve ad effettuare un "ritorno a capo"
..     move.l #Texts,a4
       moveq  #4,d7
Loop:  move.l a4,d2
       bsr    print
       add.l  #20,a4
       dbra   d7,loop
...e termina.
```

Fig. 3: Il loop serve per far stampare tutte le stringhe incrementando automaticamente il puntatore.

... NEXT nel quale sia il computer a variare ogni volta i puntatori alle stringhe ed a visualizzarle, senza dover creare un programma di dimensioni eccessive. Fortunatamente il set di istruzioni del processore 68000 comprende alcuni comandi adatti allo scopo.

Cominciamo con il caso più semplice, quello in cui le stringhe abbiano tutte la stessa lunghezza (ad esempio cinque stringhe di ventun caratteri ciascuna): la prima cosa da fare è definire la routine «Print», che in questo caso è come quella precedente (vedi fig. 3), quindi bisogna creare una zona contenente le stringhe, ed infine creare il ciclo (loop).

UN ESEMPIO

Cerchiamo di analizzare con calma il listato in figura, poiché contiene diversi concetti basilari di programmazione. Nella prima riga viene posto nel registro A4 l'indirizzo della zona contenente le stringhe da visualizzare. Il corpo del loop è rappresentato da due istruzioni e da una label («Loop:»): prima viene inizializzato con il valore 4 il registro dati D7, poi viene detto al computer di eseguire il loop tramite **l'istruzione Dbra**, che per esteso significa «Decrement and Branch Always», ovvero «decrementa e salta sempre» (nel nostro caso all'etichetta «Loop»).

Cosa fa questa istruzione? Quando usiamo un comando del tipo **Dbra Dn,Label** (dove **Dn** è un generico registro dati da D0 a D7 e **Label** è una locazione in memoria espressa tramite una label) il computer esegue le seguenti operazioni:

1) decrementa di uno il valore del registro Dn;
2) controlla se quel registro ha raggiunto il valore −1
3) se il registro ha valore diverso da −1, salta alla locazione di memoria indicata da «Label».

Nella terza riga l'indirizzo contenuto nel registro A4 è copiato nel registro indirizzi D2; nella quarta riga incontriamo invece un'istruzione nuova: **Bsr**, che significa «Branch to SubRoutine», ovvero «salta alla subroutine». L'istruzione Bsr è diversa da Jsr poiché occupa meno memoria e, grazie ad essa, si possono scrivere programmi «rilocabili», ovvero che non dipendono da una determinata locazione di memoria ma che possono essere caricati in memoria in qualsiasi punto.

Saltando alla routine «Print», viene stampata la stringa a cui punta in quel momento il registro indirizzi A4.

Nella quinta riga si aumenta il contenuto del registro A4 di 20 byte, ovvero della lunghezza (fissa) delle nostre stringhe, facendolo perciò puntare alla stringa successiva.

L'istruzione Dbra decrementa il registro dati di un byte: poiché il loop termina quando D7 ha raggiunto il valore −1, per fare eseguire cinque volte il ciclo basta porre nel registro dati il valore quattro. Se le nostre righe da visualizzare fossero state dieci, sarebbe bastato porre in D7 il valore 9.

Consideriamo ora il caso in cui, per qualsiasi ragione, non si conosca il numero esatto di righe da visualizzare ma soltanto la loro lunghezza (costante, ad esempio 21 caratteri). Possiamo allora impostare il programma nel modo seguente: considerando di avere alla label «Texts» le righe da visualizzare (fig. 4) e terminando la serie di stringhe con il valore −1, o con il valore esadecimale $FF.

Questo ciclo è profondamente diverso da quello precedente, anche dal punto di vista concettuale: in questo caso noi diciamo al computer di controllare se la riga successiva inizia con un −1; se questa condizione non è verificata, il ciclo continua (poiché significa che c'è ancora qualche altra riga da stampare), altrimenti il ciclo è terminato.

Infatti nella quinta riga, dopo aver fatto puntare il registro indirizzi A4 all'inizio della nuova linea, confrontiamo il byte contenuto in questo indirizzo con il valore −1: se il confronto ha esito negativo (se cioè il valore contenuto nel byte puntato dal registro indirizzi A4 è diverso da −1) allora il ciclo ricomincia, grazie all'**istruzione Bne** («Branch if Not Equal», ovvero «salta se è diverso»); se invece i valori sono identici il ciclo ha termine.

Per eseguire il confronto viene utilizzata l'**istruzione Cmp** ovvero «CoMPare», che significa «compara, confronta». Essendo questa istruzione (e quelle simili, come **Cmpa** e **Cmpi**) di fondamentale importanza, prima di proseguire esaminiamone la sintassi.

```
;Qui ci sono le stringhe da stampare...

Texts: dc.b    "Programma fatto da ",10
       dc.b    "-> Mickey Mouse <-",10
       dc.b    "    Disneyland    ",10
;              ...altre righe...
       dc.b    -1

;E qui la parte di programma che stampa..
       move.l #Texts,a4
Loop:  move.l a4,d2
       bsr    print
       add.l  #20,a4
       cmp.b  #-1,(a4)
       bne    Loop
;...e termina.
```

Fig. 4: Questo approccio si adotta quando non si conosce in precedenza il numero esatto di righe da stampare.

LO STATUS REGISTER

Il comando Cmp serve a confrontare una determinata locazione di memoria, o un registro, con un'altra locazione o registro, riferendosi ad esso in modo diretto o in base alle regole valide per gli indirizzamenti indiretti (ARI), esaminati nella seconda puntata di questo corso.

La sintassi è generalmente **cmp S,D** dove **S** è il primo termine di confronto (sorgente) e **D** è il secondo termine di confronto (destinazione). L'operazione effettuata è

```
                    REGISTRO DI STATO
BIT #                  DESCRIZIONE

0 \                   Carry     (C)
1  |                  Overflow  (O)
2  |                  Zero      (Z)
3  | User Byte        Negativo  (N)
4  |                  Extended  (X)
5  |                   NON USATO
6  |                      "
7 /                       "
                      Significato
8 \                   I0 \
9  |                  I1  | Maschera per gli Interrupts
10 |                  I2 /
11 |                   NON USATO
12 | System Byte          "
13 |                  Supervisore (S)
14 |                   NON USATO
15 /                  Trace      (T)
```

Fig. 5: Il registro di stato (status register) ed il significato dei sedici bit che lo compongono.

sempre R = D − S: a seconda che il risultato (R) sia maggiore, minore od uguale a zero, saranno settati (cioè posti ad uno) determinati bit in un registro a sedici bit, chiamato **Status Register** (registro di stato).

Come si può vedere in figura 5, il registro di stato (SR) è diviso nei due byte (infatti 16 bit = 2 byte): un byte di «sistema» ed un byte «utente». Al primo si accede infatti quando si vuole entrare in **modo supervisore** (generalmente riservato al sistema operativo), quando si vuole eseguire un controllo passo-passo del programma (**modo Trace**), e quando si vogliono abilitare o disabilitare determinati «livelli» di interrupt (un argomento che affronteremo più avanti nel nostro corso).

Al secondo byte, quello utente, si può accedere (oltre che direttamente, ovvero scrivendoci con un'istruzione **Move**) tramite il comando Cmp. Infatti, se R è negativo (ovvero se D è minore di S) verrà attivato il **flag N** (cioè verrà posto ad uno il contenuto di quel bit); se R è positivo (cioè se D è maggiore di S) verrà settato il **flag O**; se R è uguale a zero (cioè se D e S sono uguali) verrà attivato il **flag Z**; infine se durante la sottrazione si sarà verificato un riporto, verrà attivato anche il **flag C**.

Nota: i byte S e D non subiscono variazioni durante un'operazione di confronto.

Effettuata la comparazione, si potrà procedere ad «interrogare» i diversi flag tramite i salti condizionati. Si useranno perciò le **istruzioni Beq** (ovvero «Branch If Equal», salta se S e D sono uguali, ovvero se il bit Z è uguale a 1), e **Dbeq** («Decrement and Branch if Equal»). La figura 6 contiene una tabella di tutti gli operatori possibili per i possibili salti condizionati.

Fig. 6: Tutti i possibili tipi di salti condizionati ed i flag del registro di stato che ne influenzano il comportamento.

Tabella riassuntiva dei salti condizionati

	ABBREVIAZIONE	SIGNIFICATO	SPIEGAZIONE
	CC	Carry Clear	C = 0
	CS	Carry Set	C = 1
	EQ	Equal	Z = 1
	GE	Greater or Equal	>=
***	GT	Greater Than	>
	HI	Higher	>
***	LE	Less or Equal	<=
	LS	Less or Same	<=
***	LT	Less Than	<
	MI	Minus	-
	NE	Not Equal	<>
	PL	Plus	+
***	VC	Overflow Clear	V = 0
***	VS	Overflow Set	V = 1

È importante notare che esistono operatori che tengono conto di diversi flag, ed altri (quelli contrassegnati con gli asterischi) che tengono conto anche dei numeri dotati di segno. Questo tipo di salto funziona correttamente solo se immediatamente vicino ad un'istruzione Cmp. La figura 7 contiene alcuni esempi.

SALTI CONDIZIONATI

L'esempio 1 evidenzia come viene usata l'**istruzione Bcc** («Branch on Condition Code», cioè «Salta a seconda della condizione»): confrontiamo D0 con il numero 10 (**Cmp.w #10,D0**, ma si poteva anche scrivere **Cmpi.w #10,D0**, dato che nella sintassi il comando Cmp è identico all'istruzione Add, per cui esistono Cmp, Cmpi e Cmpa) e, se è diverso, saltiamo a Loop (ma si poteva anche ricorrere a «**Bls Loop**» o ad altro).

Nell'esempio 2 inizializziamo il registro D0 con il valore decimale 40 e poi lo decrementiamo sino a quando

```
1)   moveq    #10,d0        2)    moveq    #40,d0
Loop:  ...                  Loop:  ...
     addq.w   #1,d0               subq.w   #1,d0
     cmp.w    #40,d0              cmp.w    #40,d0
     bne      loop                bne      loop

3)   move.b   #$0a,d0       4)    move.b   #$0a,d0
     move.b   #$f0,d1             move.b   #$f0,d1
     move.b   #$10,d2             move.b   #$10,d2
     cmp.b    d0,d1               cmp.b    d0,d1
     beq      l1                  add.b    d1,d2
     add.b    d1,d2               beq      l1
     rts                          rts
l1:  move.b   #$20,d0       l1:   move.b   #$20,d0
     rts                          rts
```

Fig. 7: Alcuni esempi di salti condizionati basati sul risultato di confronti tra i contenuti dei registri.

non ha raggiunto il valore 10, utilizzando l'**istruzione Bge** (che non termina il ciclo sino a quando il registro dati D0 è superiore a 10).

Gli esempi 3 e 4 sono molto simili fra loro: nell'esempio 4 sono state infatti scambiate due righe rispetto al precedente. Vediamo cosa accade.

Nell'esempio 3, dopo aver posto dei valori in D0, D1 e D2 controlliamo che D0 e D1 siano uguali; poiché la condizione non è verificata, sommiamo D1 a D2 e terminiamo.

Nell'esempio 4, dopo aver posto dei valori in D0, D1 e D2, controlliamo che D0 e D1 siano uguali, sommiamo D1 a D2 ed eseguiamo un salto se è verificata la condizione. Se però assembliamo e lanciamo questo mini-programma, il risultato sarà di avere, nel registro dati D0, il valore 20, segnale che indica l'avvenuto salto imposto dall'**istruzione Beq L1**. Perché?

A differenza dell'esempio 3, non possiamo verificare la condizione espressa dall'**istruzione Cmp.b D0, D1**, poiché tra essa e l'**istruzione Beq L1** si interpone l'**istruzione Add.b D1, D2.** I valori di D1 e D2 sono stati impostati in modo che una loro somma dia come risultato il valore zero: infatti $F0 + $10 = $100 ma, poiché la somma è estesa al primo byte del registro dati, come risultato dell'operazione il primo byte di D2 conterrà il valore zero. Perciò, una volta effettuata l'operazione, il computer setterà ad uno il flag Z, e risulterà verificata la condizione del salto (Beq, branch if equal con z = 1, ricordate ?).

```
print: move.l  #Texts,a5        ;indirizzo del testo in a5
loop2: move.l  a5,a4            ;metti a5 in a4
       moveq   #0,d3            ;a zero d3 (la lunghezza)
loop1: addi.w  #1,d3            ;lunghezza in d3
       cmpi.b  #10,(a4)+        ;a4 punta al valore 10?
       bne     loop1            ;no , aumenta la lunghezza della
                                ;stringa
       move.l  handle,d1        ;si', metti l' handle in d1
       move.l  a5,d2            ;indirizzo del testo in d2
       move.l  DosBase,a6       ;indirizzo dos.library in a6
       jsr     write(a6)        ;esegui la funzione write
       move.l  a4,a5            ;metti a4 in a5 (punta alla nuova
                                ;stringa)
       cmp.b   #-1,(a4)         ;a4 punta al valore -1?
       bne     loop2            ;no , continua a visualizzare
       rts                      ;si', esci.
Handle:   dc.l 0                ;spazio per l' handle di output
Texts:    dc.b " ciao a tutti da ",10      ;il testo da visualizzare
          dc.b "  a m i g a b y t e !!!",10
          dc.b -1               ;il testo e' terminato
```

Fig. 8: Un sorgente dimostrativo che esemplifica come stampare a video un gruppo di stringhe di lunghezza qualsiasi.

Da ciò derivano due regole molto importanti: 1) imporre sempre un salto condizionato subito dopo un confronto; 2) evitare, nel codice, l'inserimento di confronti inutili dopo operazioni di somma e riporto o simili che possano in qualche modo far cambiare i bit nel registro di stato.

Sarebbe stato perfettamente inutile, nell'esempio 4, porre:

add.b D1,D2
cmp.b #0,D2
beq L1

poiché l'operatore Beq prevede già un confronto con il flag Z, che viene settato (se necessario) dopo l'istruzione **add.b D1,D2.**

È invece una saggia misura precauzionale inserire il confronto **cmp.b #0,D2** se tra l'addizione ed il salto ci sono altre istruzioni che potrebbero far variare lo Status Register. Ad esempio:

add.b D1,D2
move.1 #$40000,A0
subq.1 A1,A0
...
cmp.b #0,D2
beq L1

IL SORGENTE

Per finire, esaminiamo la routine «Print» del sorgente dimostrativo di questa puntata in figura 8: questo programma, da compilare come di consueto con il **Seka Assembler**, stampa una serie di linee di lunghezza qualsiaisi sullo schermo.

Innanzitutto, memorizziamo l'indirizzo del testo nel registro A5, quindi poniamo questo indirizzo in A4: in questo modo possiamo incrementare A4 per vedere se ha raggiunto il valore 10 (ritorno carrello), il che significa che è terminata la prima riga da visualizzare (si suppone in questo caso che il testo abbia almeno una riga!).

Poiché la routine Write del sistema operativo richiede che in D3 ci sia la lunghezza della stringa da visualizzare, incrementiamo direttamente questo registro (ricordandoci di riazzerarlo all'inizio di ogni nuovo loop) sino a quando il byte puntato da A4 non conterrà il valore 10.

Terminato questo primo loop, poniamo in D2 l'indirizzo della stringa da visualizzare (contenuto in A5), mettiamo l'handle di output in D1, e saltiamo alla routine Write.

Visualizzata la stringa, esaminiamo il contenuto dell'indirizzo nel registro A4, che sappiamo puntare alla nuova stringa: se il valore del byte puntato da A4 è −1, siamo giunti alla fine del testo; altrimenti, c'è ancora qualcosa da visualizzare. È sufficiente quindi mettere il valore di A4 in A5 e ricominciare.

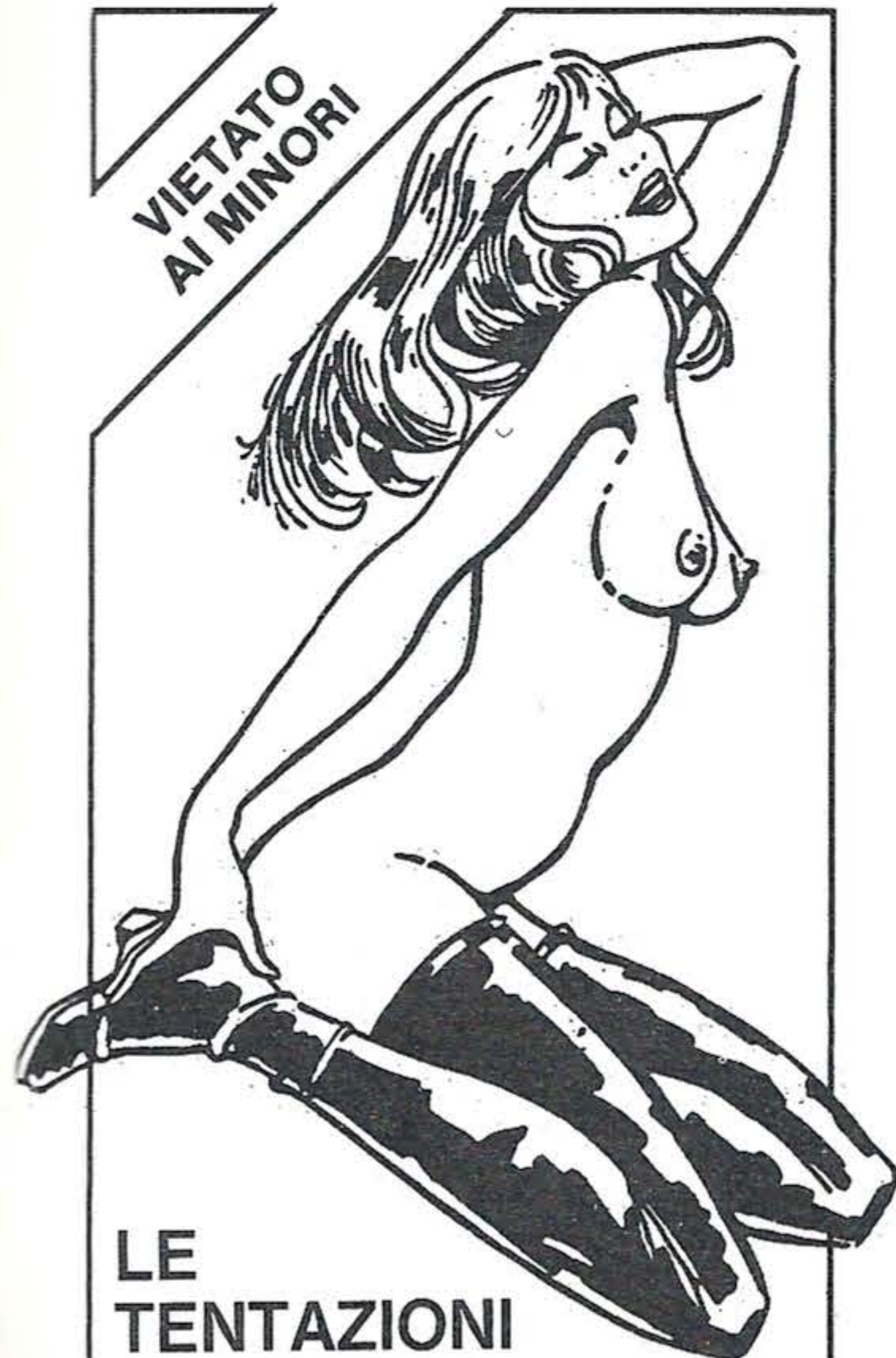
Per quanto riguarda il testo da visualizzare, è ammessa

qualsiasi sequenza di tasti (come con il comando «Echo» di AmigaDos). L'unica cosa da tener presente è l'impossibilità di usare le virgolette (" ") all'interno della stringa stessa, poiché esse vengono considerate dall'assemblatore come delimitatore di inizio e fine stringa; l'assemblatore visualizzerebbe un messaggio di errore per una linea come la seguente:

dc.b "Mario dice "ciao" a Pippo", 10
mentre accetterebbe
dc.b "Mario dice 'ciao' a Pippo", 10
senza problemi.

Nella figura 9 sono mostrati i comandi necessari per assemblare il listato proposto con il Seka: ricordate solamente di non eseguire il programma dall'interno dell'assemblatore, ma di uscirne prima con il comando «!» e di lanciarlo digitandone il nome da CLI, come qualsiasi altro file eseguibile.

□

Fig. 9: I comandi necessari per assemblare e salvare il sorgente usando il Seka Assembler.

LE TENTAZIONI DI AMIGA
solo per adulti

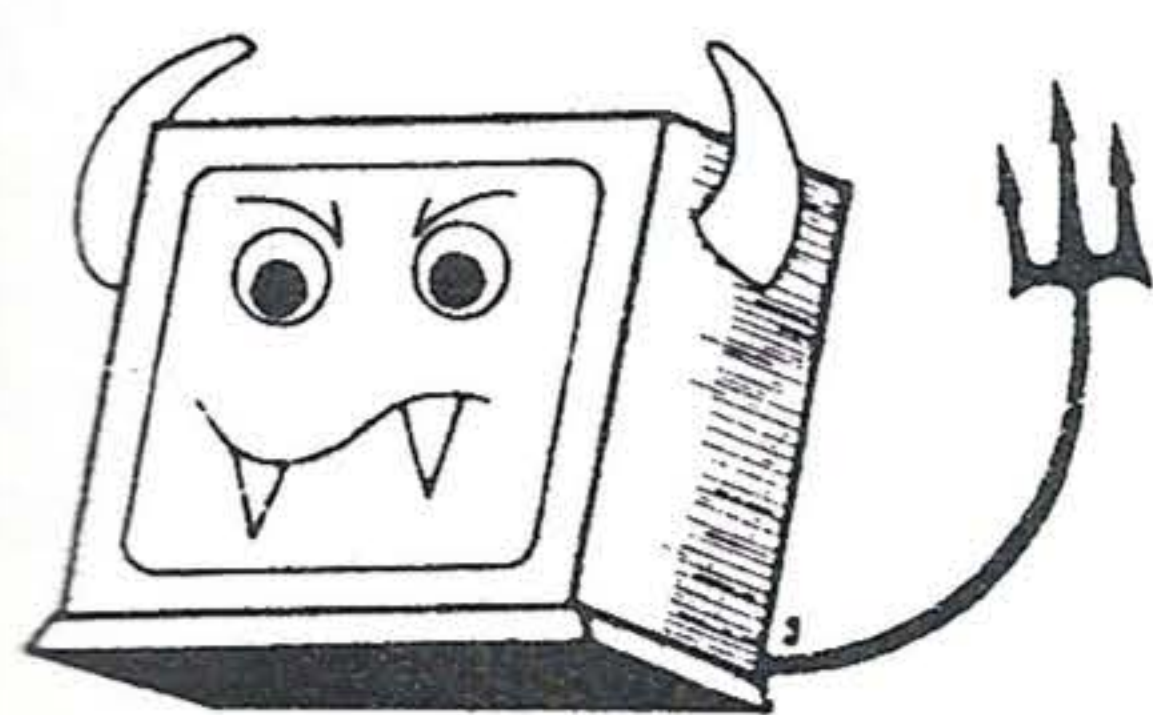
■ **AMI PORNO SHOCK**

Due dischetti con le immagini più hard mai viste sul tuo computer e un'animazione che metterà a dura prova il tuo joystick!

Lire 25mila

■ **PORNO FILM**

È il conosciutissimo (per chi ce l'ha...) AmigaByte PD7: un dischetto eccezionale con tre film. Julie, Bridget e Stacy i tre titoli. I primi due di animazioni, il terzo un favoloso slideshow con definizione e dettagli che stupiscono.

Lire 10mila

Per ricevere AmiPornoShock oppure PornoFilm basta inviare vaglia postale ordinario ad AmigaByte, C.so Vitt. Emanuele 15, Milano 20122. Specifica sul vaglia stesso la tua richiesta (Shock oppure Film) e naturalmente il tuo indirizzo. Per un recapito più rapido aggiungi lire 3mila e chiedi spedizione espresso!

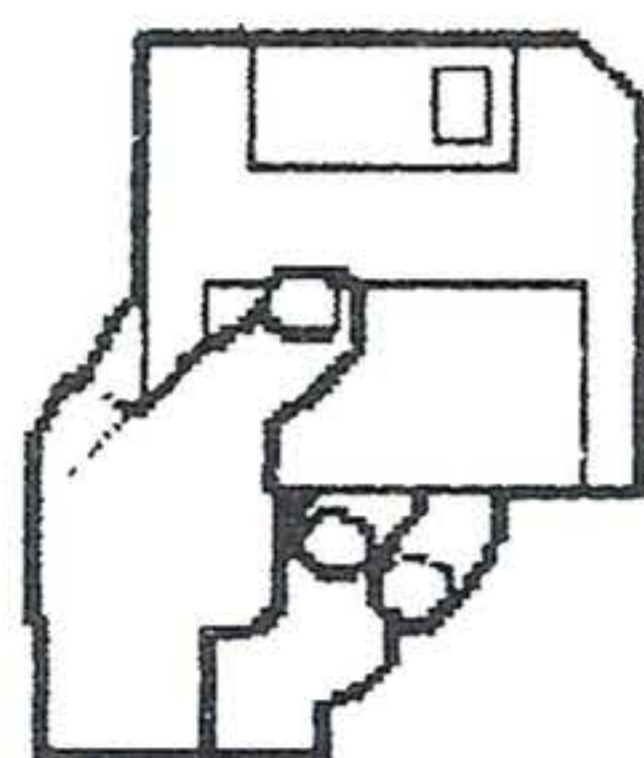
nuovissimo CATALOGO

SOFTWARE PUBBLICO DOMINIO

* *Il catalogo viene continuamente aggiornato con i nuovi arrivi!!!*

CENTINAIA DI PROGRAMMI

UTILITY
GIOCHI
LINGUAGGI
GRAFICA
COMUNICAZIONE
MUSICA

IL MEGLIO DEL PD
e in più
LIBRERIA COMPLETA FISH DISK 1 - 460

*** SU DISCO ***

Per ricevere il catalogo su disco invia vaglia postale ordinario di lire 10.000 a AmigaByte C.so Vitt. Emanuele 15 20122 Milano

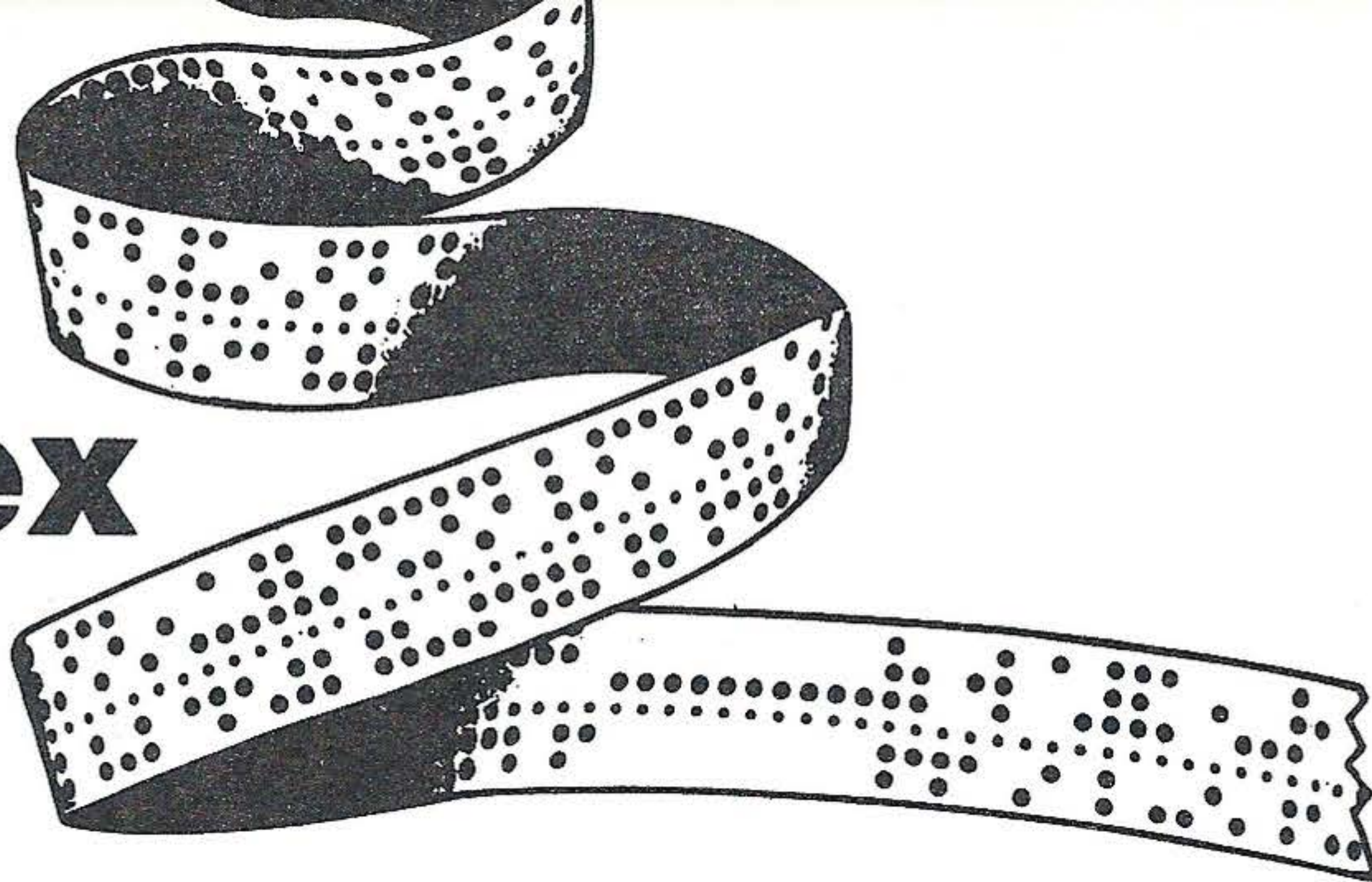
PER UN RECAPITO PIÙ RAPIDO aggiungi L. 3.000 e richiedi SPEDIZIONE ESPRESSO

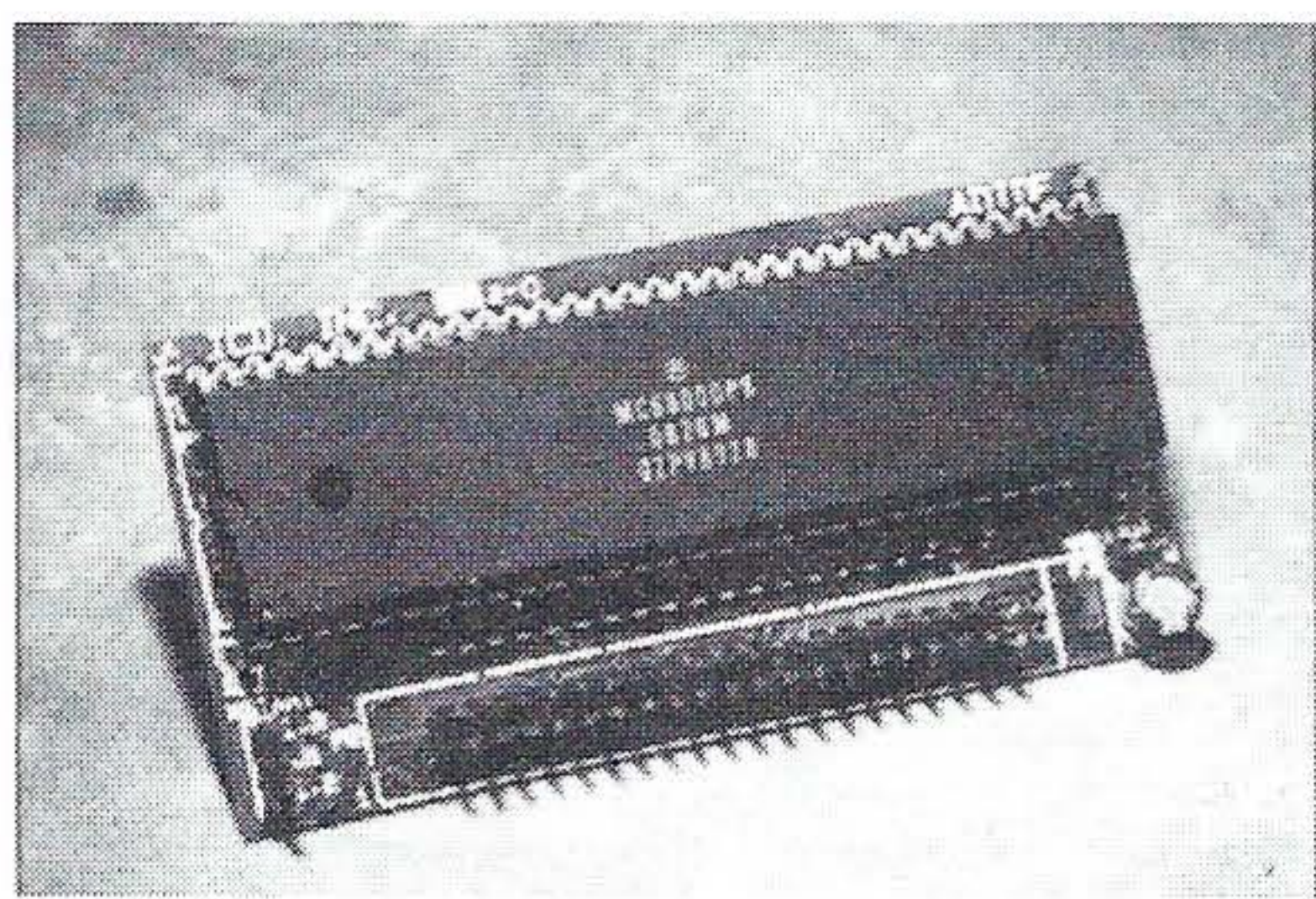
Sì,
anche tu puoi collaborare ad Amiga Byte!

Con articoli, megagame, idee...

La redazione è a tua disposizione per vagliare ogni lavoro

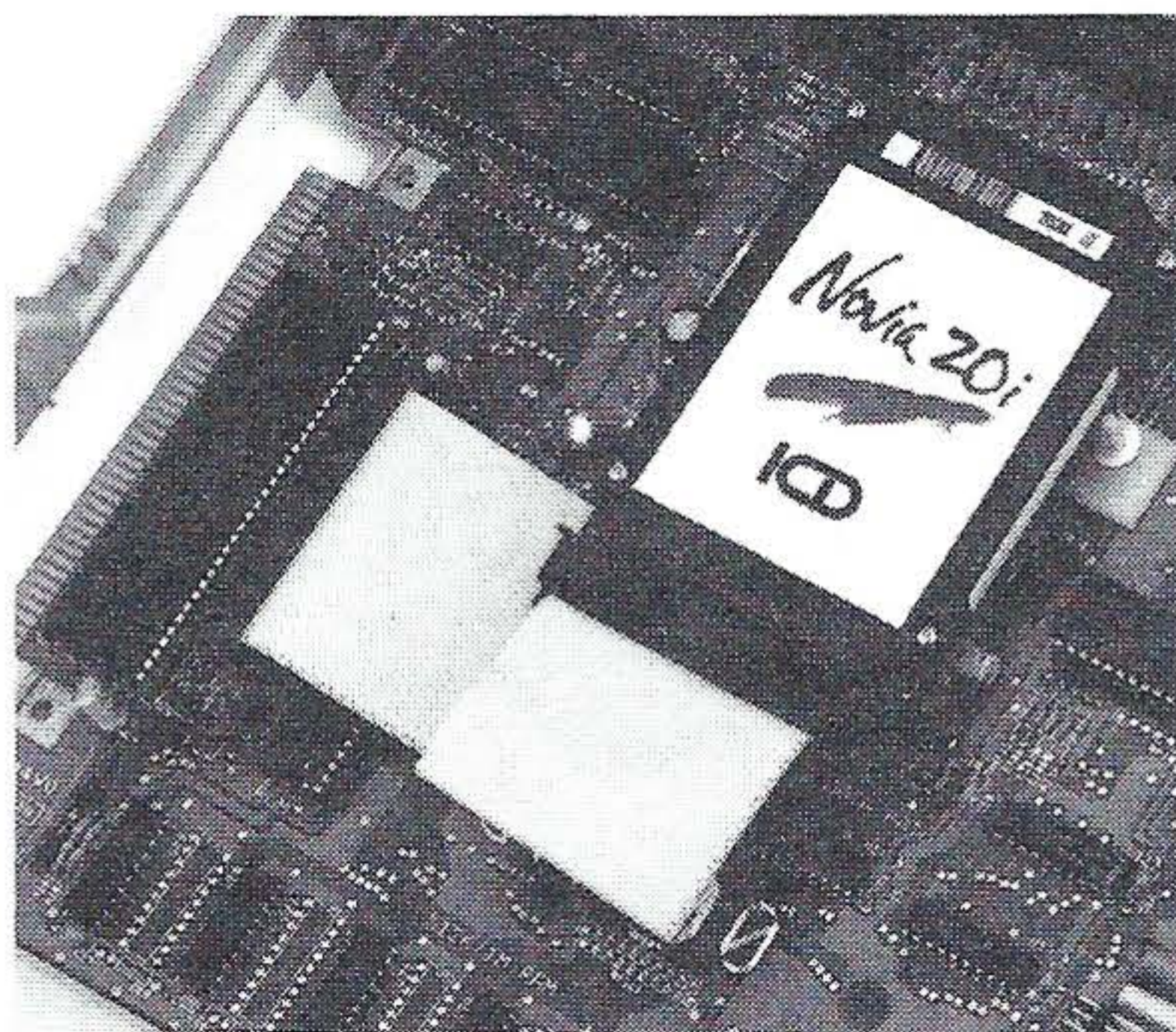
Invia una scaletta di quello che pensi di poter fare o un dischetto con le tue creazioni

Spedisci ad AMIGABYTE c.so Vitt. Emanuele 15 20122 Milano

ICD: Micro accessori per Amiga

La palma della scheda acceleratrice più piccola spetta sicuramente alla microscopica «**AdSpeed**», prodotta dalla ICD (1220 Rock Str., Rockford, IL 61101, USA): le sue dimensioni sono di poco superiori a quelle della CPU 68000 presente all'interno di tutti

gli Amiga (3000 escluso), e la sua installazione risulta quindi semplice ed immediata.

L'«AdSpeed» consta di un chip 68000 a 14 Mhz (velocità all'incirca doppia di quella di clock del normale 68000 montato nel computer) e di una minuscola schedina sulla quale sono presenti 32K di ram cache. Il prezzo ridotto (circa 250 $) e la semplicità del prodotto giustificano le prestazioni certamente non paragonabili a quelle di schede acceleratrici basate su processori a 32 bit. In ogni caso, si tratta di un'alternativa interessante soprattutto per i possessori di Amiga 500 che desiderano un incremento di prestazioni senza pretendere velocità da capogiro.

Analogamente microscopico è il controller per hard disk ICD «**AdIde**». I controller IDE (abbreviazione di Integrated Drive Electronics) si differenziano da quelli standard in quanto vengono montati direttamente sull'hard disk e non occupano slot. L'«AdIde» si installa nello zoccolo della CPU, essendo dotato di un processore 68000 proprio, e consente il collegamento di hard disk formato AT-Bus. Se usato con un minuscolo hard disk da 2"1/2, come quelli ora in voga normalmente usati sui PC portatili (i cosiddetti *notebook*), questo controller rende possibile installare un disco rigido anche *internamente* ad Amiga 500. Prezzo del mini-controller: circa 160$. La ICD commercializza anche il pacchetto completo (controller + micro hard disk da 20 MB) con il nome «**Novia 20i**» a meno di 600$.

Ancora più interessante è la scheda «**Flicker Free Video**», sempre della ICD, venduta negli USA per cifre che si aggirano sui 350$: a differenza di accessori analoghi, come il «Flicker Fixer» della Microway, usati per rimuovere l'inconveniente dello sfarfallio delle immagini in alta risoluzione, il prodotto della ICD non

richiede di essere installato nello slot video (presente solo su Amiga 2000 e 3000) e può quindi essere usato tranquillamente anche sui modelli 500. L'installazione avviene rimuovendo il chip video Denise dalla scheda madre di Amiga, montandolo nella «Flicker Free Video», ed inserendo quest'ultima nel computer nello zoccolo che ospitava il chip. A detta della casa produttrice, la scheda è assolutamente trasparente a qualsiasi programma e non interferisce con il normale comportamento del segnale video di Amiga, permettendo quindi l'utilizzo di genlock ed altre periferiche analoghe.

La scheda si adatta automaticamente allo standard video dell'Amiga in cui è installata (NTSC o PAL) e supporta anche l'overscan (con una risoluzione massima di 832 × 568 pixel). Per rimuovere lo sfarfallio, è comunque indispensabile l'utilizzo del «Flicker Free Video» in abbinamento ad un monitor multisync (come il Commodore 1950).

Se il 500 non basta più

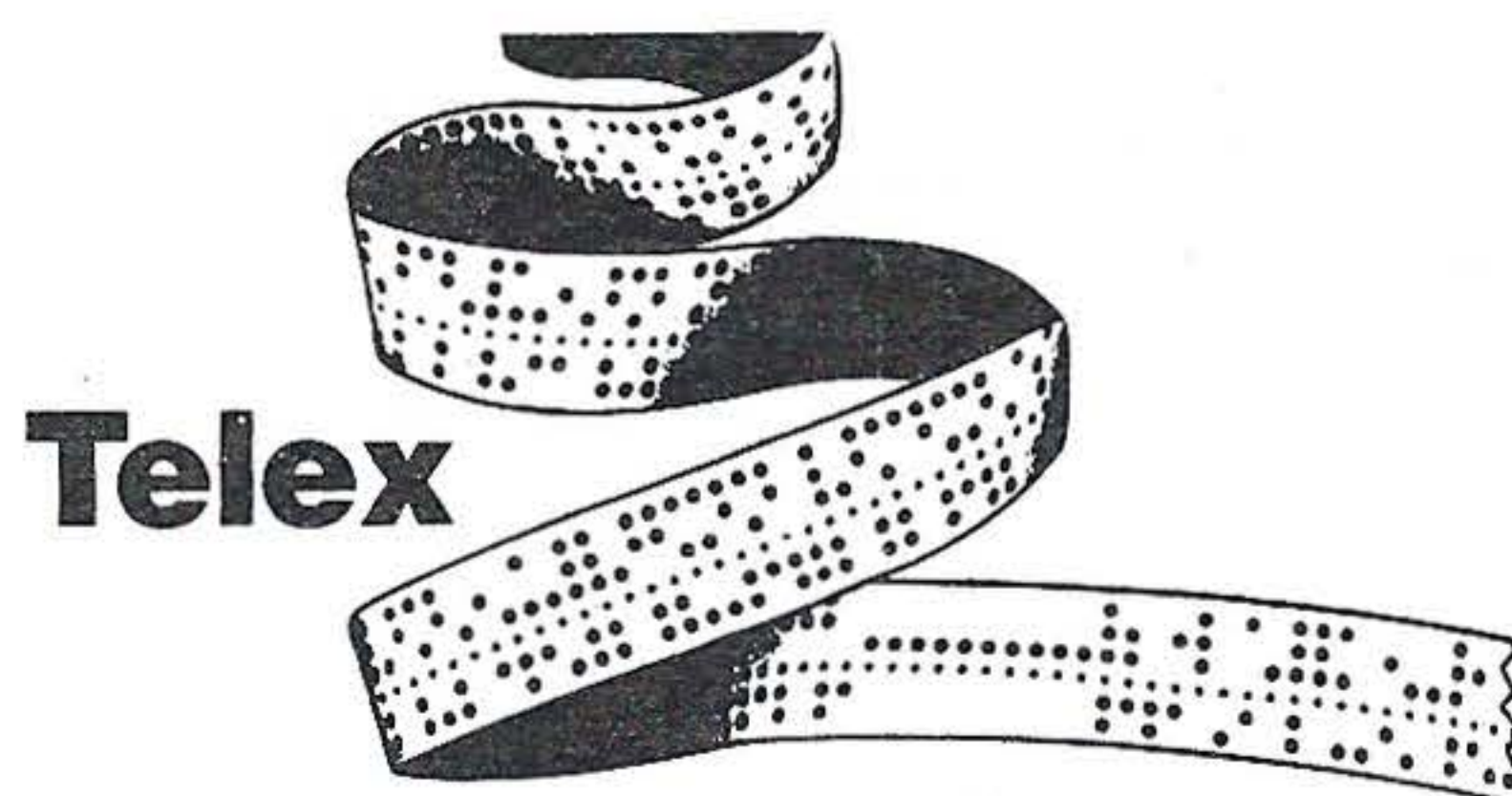
Espandere un Amiga 500 è sempre un problema: lo spazio all'interno del computer è ridottissimo, e l'assenza di slot impedisce l'uso di tutte le numerose schede ed accessori disponibili per Amiga 2000 e derivati.

La soluzione proposta dall'americana California Access (130 Knowles Drive, Los Gatos, CA 95030, USA) si chiama «**Bodega Bay**» e consiste in un box esterno, collegabile ad Amiga 500 per mezzo del bus di espansione laterale, che aumenta lo spazio a disposizione per l'installazione di add-on. Esso comprende quattro slot da 100 pin (tre dei quali adatti anche ad accogliere schede di tipo XT/AT), equivalenti a quelli presenti in Amiga 2000 e 3000, consentendo quindi anche di installare una scheda Janus/BridgeBoard per la compatibilità MsDos. Per facilitare l'uso di quest'ultima, il «Bodega Bay» dispone di due alloggiamenti frontali per accogliere drive aggiuntivi da 3" o da 5"1/4

(come quello fornito con la scheda Janus). Il box è dotato di un potente alimentatore interno da 200 watt, raffreddato con una ventola, che si sostituisce quindi a quello esterno normalmente usato da Amiga 500.

Due sono principalmente i problemi derivanti dall'uso del «Bodega Bay» (oltre che il prezzo di quasi 400 $): il primo è l'occupazione dello slot laterale di Amiga, che impedisce quindi l'aggiunta di un hard disk esterno come l'A590, e costringe all'acquisto di hard disk interni per Amiga 2000, con relativo controller.

A questo proposito, la California Access sconsiglia l'uso dei cosiddetti *hard card* (hard disk montati direttamente sulla scheda controller) poiché, essendo gli slot disposti verticalmente (come quelli di Amiga 3000) sul lato interno sinistro del box, le schede devono essere inserite orizzontalmente ed i drive montati su di esse risulterebbero capovolti, posizione che in alcuni casi non favorisce il corretto movimento delle testine.

Il «Bodega Bay» inoltre non è purtroppo provvisto né di slot per un coprocessore né di slot video analoghi a quelli di Amiga 2000/3000, il che vieta l'utilizzo di tutte quelle periferiche (Genlock, Frame Buffers, schede anti flicker e l'ormai leggendario «Video Toaster») che richiedono di essere installate in questi slot.

La California Access ha tuttavia annunciato la futura possibilità di upgrade per aggiungere anche queste funzionalità.

Nonostante le limitazioni, il «Bodega Bay» è tuttavia un prodotto molto interessante, specialmente per i numerosi possessori di Amiga 500 che desiderano espandere al massimo il proprio computer senza dover effettuare un cambio completo con un modello più potente. □

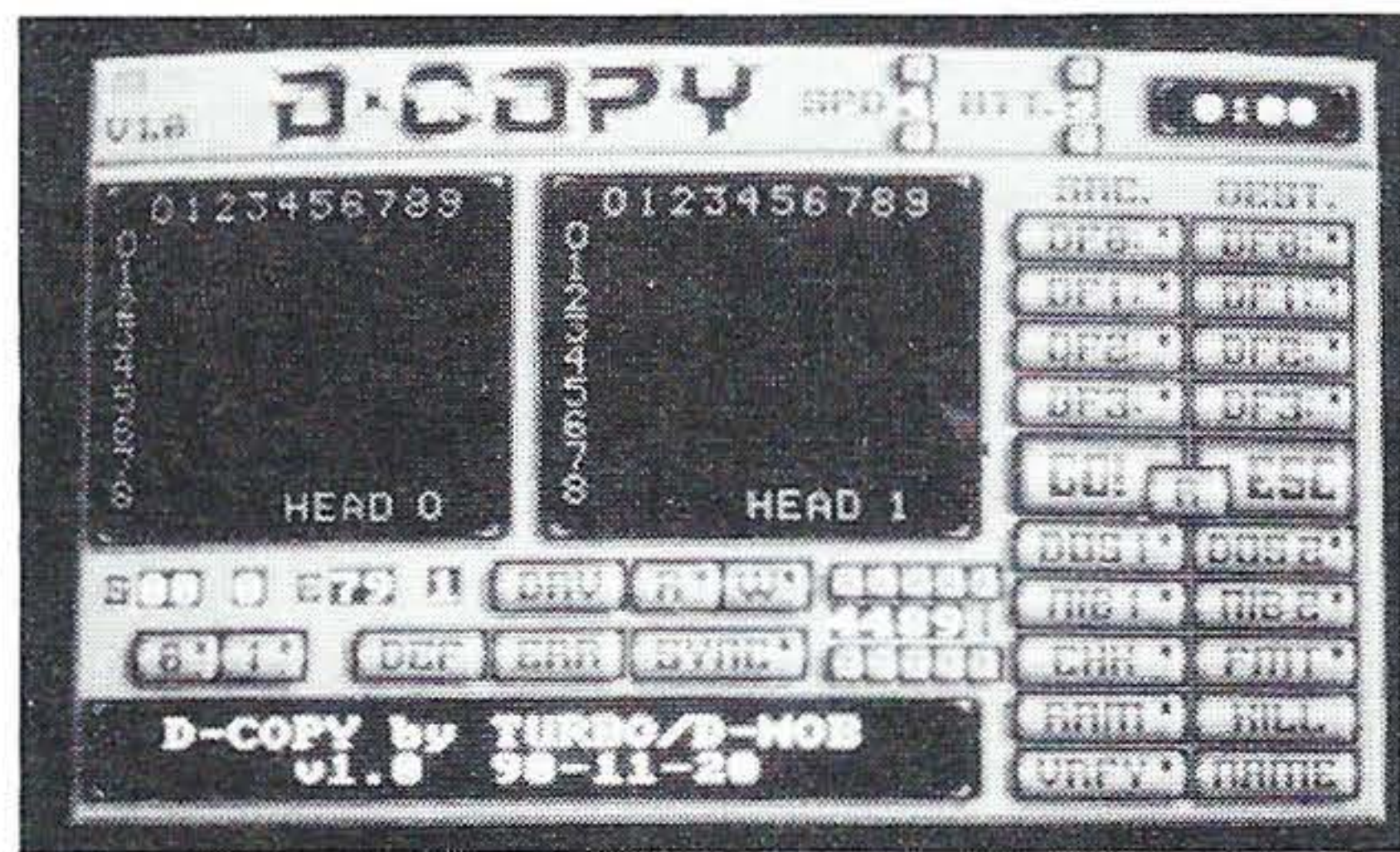
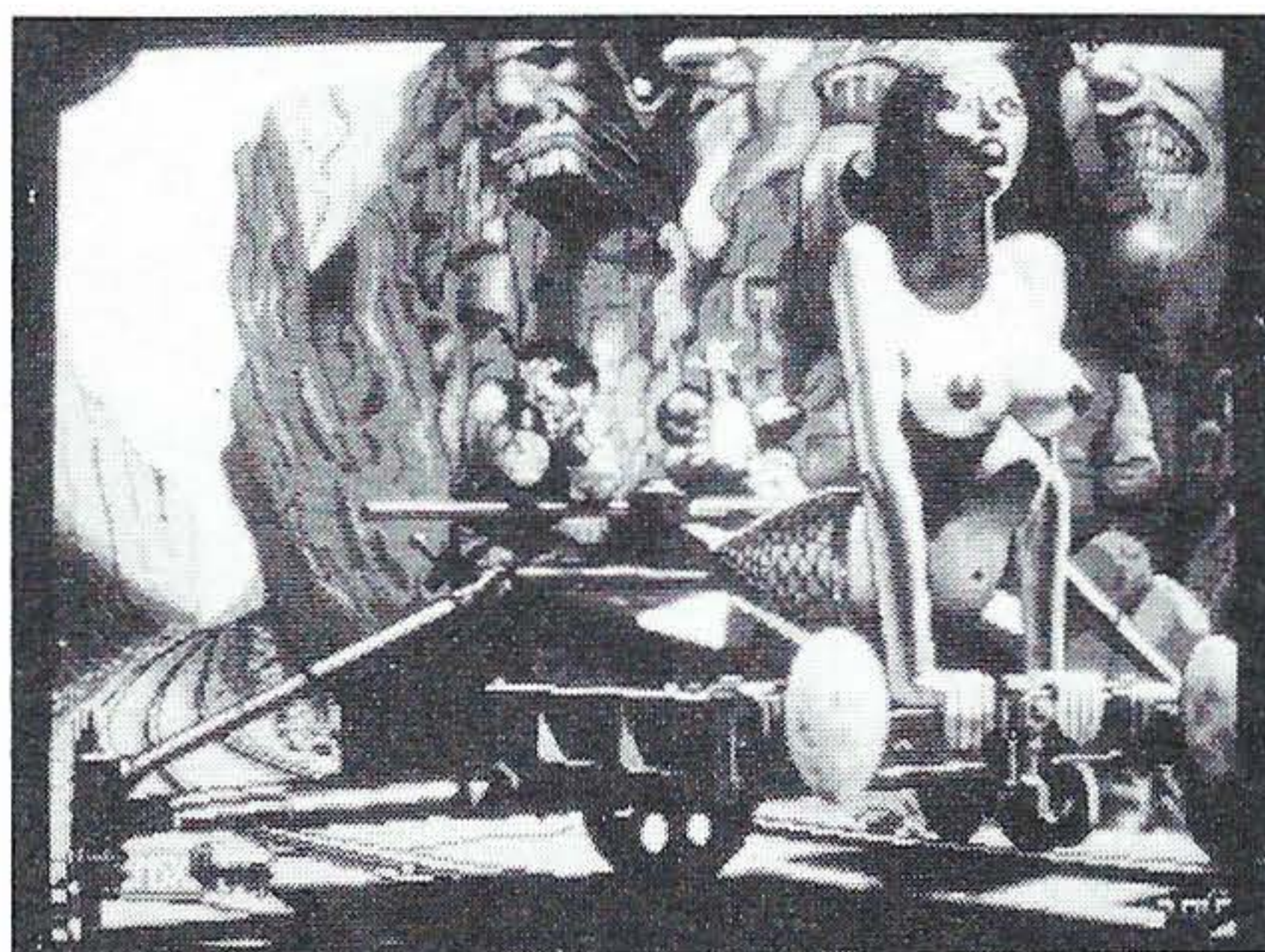
in edicola fascicolo speciale!

The best of the PD

Abbiamo scelto per voi, e continueremo a farlo, i migliori programmi fra quelli di Pubblico Dominio esistenti che, per ragioni di spazio, non possono essere inseriti nel dischetto allegato ad Amiga Byte.

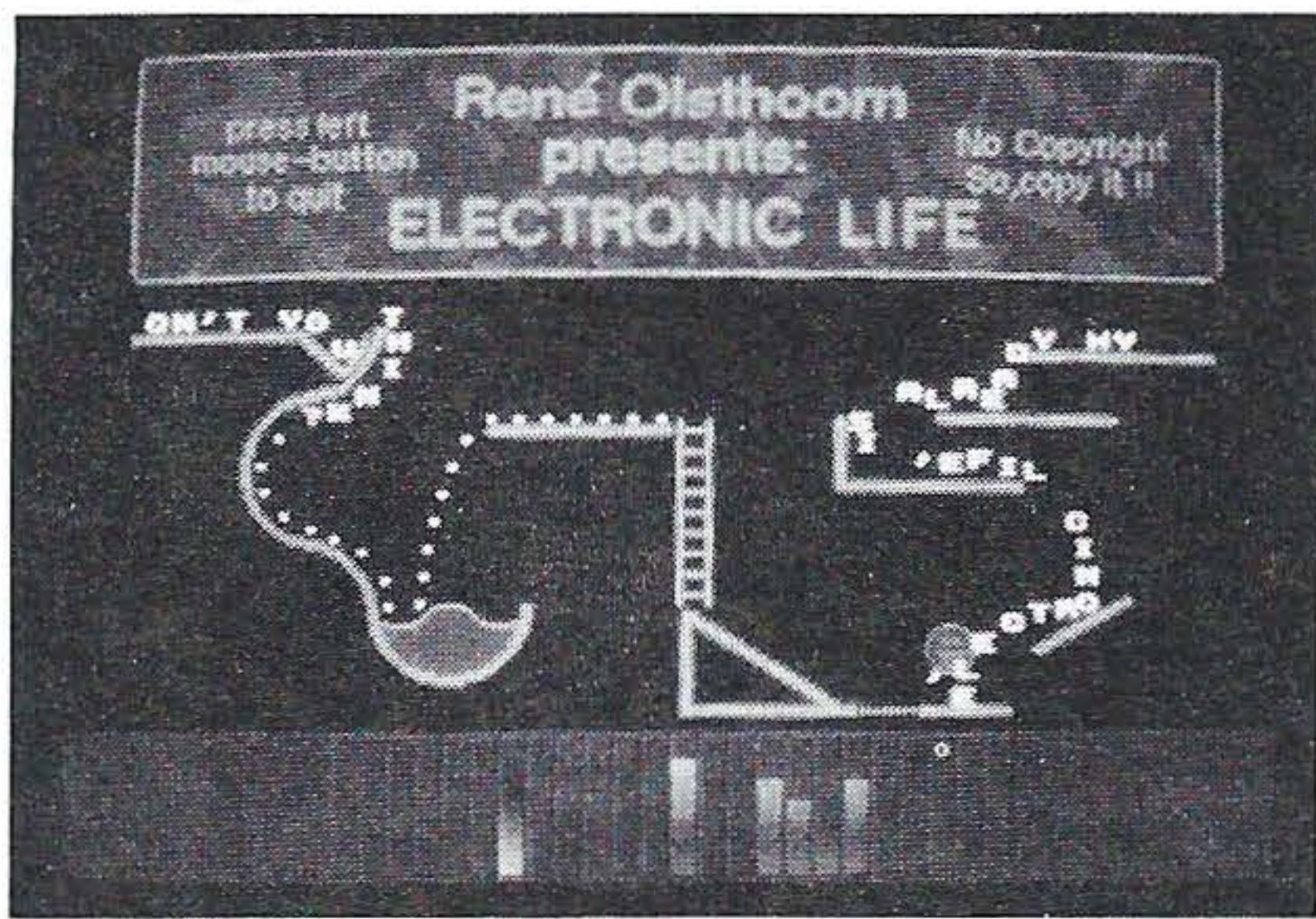
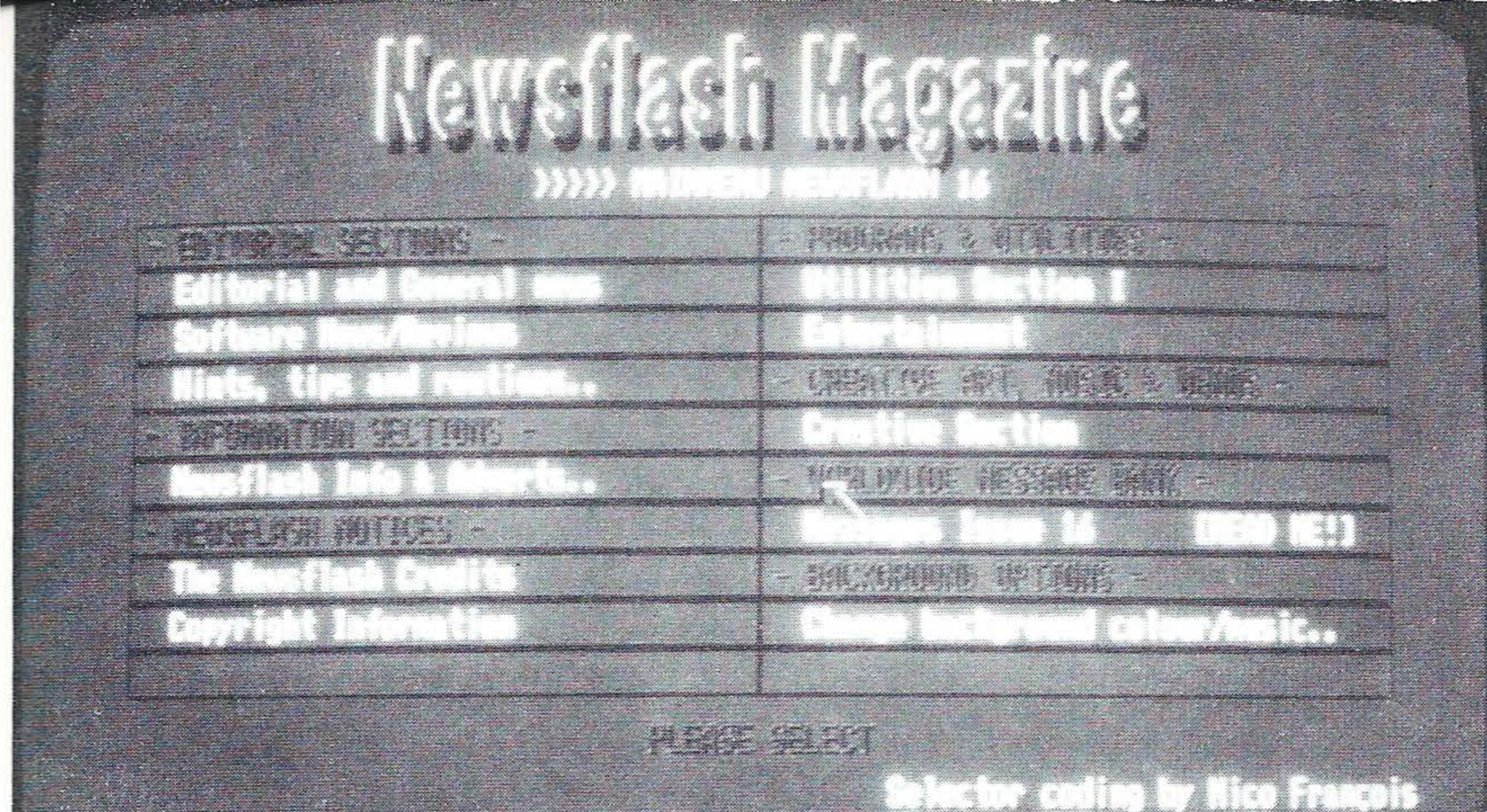
di EMANUELE SCRIBANTI

UGA SOFTWARE

Il nome UGA è l'abbreviazione di United Graphic Artistis, un gruppo di artisti, musiciti e programmatori Amiga europei che, sotto la guida di Ron Fonteine, organizzatore e coordinatore principale delle attività commerciali del gruppo, è diventato da tempo un marchio familiare a tutti coloro che operano nel settore del software di pubblico dominio per Amiga.
Una tra le iniziative UGA più note riguarda la creazione di una serie di dischetti di pubblico dominio, diffusa in tutta Europa e particolarmente apprezzata dagli utenti e dai programmatori Amiga più attenti.
I dischi della serie UGA raccolgono infatti una quantità enorme di software PD di produzione prevalentemente europea, ordinati per argomento. A differenza di altre raccolte di software PD, come ad esempio i dischi della libreria di Fred Fish, i dischetti UGA sono autobootanti e possono essere usati senza bisogno di WorkBench o di altri programmi accessori.
Il materiale UGA viene distribuito secondo

una formula denominata «LicenceWare»: i dischetti appartenenti alla collezione UGA PD possono infatti essere liberamente copiati e distribuiti, ma non possono essere venduti o forniti dietro pagamento di un rimborso spese (nemmeno per coprire i costi di duplicazione e del supporto magnetico), poiché contengono materiale esclusivo i cui diritti sono detenuti esclusivamente da UGA.
La ragione di questa limitazione risiede nel fatto che UGA compensa gli autori dei programmi con una percentuale sui dischetti venduti; pertanto solo i dischetti originali, riconoscibili per la loro caratteristica etichetta recante il marchio UGA, possono essere commercializzati e fatti circolare.
La diffusione di questi dischi è affidata da UGA ad una serie di distributori, in Europa e negli Stati Uniti, che si occupano anche della vendita del software UGA commerciale

(al quale accenneremo più avanti).
AmigaByte ha stipulato con UGA un accordo per la distribuzione del suo software in esclusiva sul territorio italiano, ed è in grado di fornire ai propri lettori i dischetti della collezione UGA PD, della rivista su disco NewsFlash e delle altre iniziative software di questo intraprendente gruppo olandese.
I dischi UGA PD comprendono software normalmente non reperibile in altre collezioni di materiale freeware o shareware: nel catalogo UGA troviamo infatti 12 dischetti di brani musicali in formato Aegis Sonix, comprensivi di strumenti; 19 di musiche create con SoundTracker e NoiseTracker (o analoghe utility PD), tutte autoeseguibili; 10 di animazioni grafiche generate da Sculpt, VideoScape, DeluxePaint, FantaVision ed altri diffusi tool grafici; 8 slideshow pieni di suggestive immagini IFF, per la maggior parte inedite e create dal team di artisti UGA; 12 compilation di utility varie (antivirus, tools per il disco, editor grafici, ecc.); 19 raccolte di demo ed intro, comprendenti il meglio della produzione degli hacker europei, e via di questo passo.
Non mancano ovviamente, oltre a queste raccolte a tema, dischetti con programmi particolari che, per la loro importanza o dimensione, sono stati racchiusi in una serie denominata UGA PD Special.
Tra di essi spiccano i tool musicali SoundMonitor (UGA SP 8 e 9), Jam Cracker e Games Music Creator (UGA SP

11); quest'ultimo è particolarmente noto perché genera moduli musicali in un formato supportato dall'interprete Basic AMOS. Il disco UGA SP 14 contiene invece una serie di 35 suoni inediti campionati, da usare con qualsiasi programma musicale che richieda l'impiego di samples (ad esempio NoiseTracker).
Sempre nella serie Special troviamo due raccolte di font a colori in formato IFF, da usare con Deluxe Paint, o altri programmi di grafica e desktop video per la creazione di spettacolari scritte e logotipi (UGA SP 5 e 12).
Il catalogo del software di pubblico dominio di AmigaByte include l'elenco completo del contenuto di tutti i dischetti UGA, che possono essere richiesti presso la redazione seguendo le solite modalità (vaglia postale).

NEWSFLASH

NewsFlash è il più diffuso ed interessante disk magazine per Amiga attualmente disponibile in Europa. La periodicità è mensile: ogni numero è costituito da due dischetti contenenti utility, giochi, listati e corsi di programmazione in AMOS, C ed Assembler, brani musicali, demo grafiche e sonore, recensioni di nuovi prodotti hardware e software, suggerimenti e soluzioni per giochi ed avventure, font, immagini, animazioni e reportage sulle mostre e le manifestazioni Amiga in tutta Europa.
Tutto i programmi ed i testi (in lingua inglese) contenuti nei dischetti NewsFlash sono compressi con PowerPacker Pro, in modo da fornire ogni mese un minimo di 2 mega di software assortito. Ogni numero di NewsFlash viene redatto parzialmente in Gran Bretagna da Martyn Brown, organizzatore e manager della società inglese di distribuzione di materiale PD «17Bit Software», e per la parte restante in Olanda da Ron Fonteine, raccogliendo il meglio del software PD in circolazione in tutta Europa ed includendo materiale creato in esclusiva per NewsFlash dal team di programmatori e collaboratori UGA.
NewsFlash è il metodo migliore per essere sempre aggiornati sulle novità della scena PD europea, in quanto il software in esso contenuto in genere non ha più di un mese di vita dal momento della pubblicazione. Oltre che per i programmi, NewsFlash è particolarmente interessante per la parte

(denominata «Creative Section») riservata al materiale artistico creato dai membri UGA e dai lettori. Ogni numero comprende sempre almeno un paio di nuovi brani musicali inediti e di immagini grafiche di qualità spesso superlativa.
Molti dei più noti artisti e musicisti della scena Amiga europea contribuiscono regolarmente a NewsFlash con anteprime dei propri lavori: le immagini futuristiche ispirate a Star Trek di Tobias Richter ed i brani musicali di Alistair Brimble e Brian Postma sono solo alcuni esempi apparsi nei numeri precedenti di NewsFlash.
Per gli smanettoni, NewsFlash è una fonte inesauribile di materiale: i numerosi nuovi demo inclusi in ogni numero possono fornire (tramite l'uso di qualche programma-ripper) immagini, moduli musicali, font e samples da utilizzare nelle proprie creazioni. La presenza di sorgenti Assembler e C, comprendenti ad esempio routine di scrolling, o per il display di barre ed equalizzatori durante le riproduzioni di musiche, ed i numerosi listati per Amos rendono NewsFlash indispensabile a tutti i programmatori bisognosi di aiuto.
Il successo di NewsFlash è cresciuto proporzionalmente al suo valore: i primi cinque numeri comprendevano un solo dischetto ciascuno; attualmente, per contenere tutto il materiale disponibile mensilmente, NewsFlash viene fornito su due dischetti, e non è escluso che ben presto diventino tre.
Inizialmente diffusa in pochi esemplari, la rivista NewsFlash viene ora distribuita regolarmente in tutto il mondo: in Olanda, patria d'origine, ogni numero viene venduto direttamente nelle edicole. Analogamente a quanto accade negli Stati Uniti, dove la distribuzione è affidata alla popolare rivista americana su disco JumpDisk, in Italia è AmigaByte a distribuire NewsFlash. A partire dal prossimo numero presenteremo mensilmente il contenuto delle uscite di NewsFlash e le modalità per riceverlo.
L'elenco dei numeri arretrati disponibili (dal numero 1 al numero 18) e del software in essi contenuto è incluso nel catalogo su disco del software PD di AmigaByte.

Per ricevere i dischetti contenenti i programmi recensiti su queste pagine, basta inviare vaglia postale ordinario di Lire 10 mila ad AmigaByte, c.so Vitt. Emanuele 15, 20122 Milano, specificando il codice del disco (esempio: Amiga Byte PD 2) ed i propri dati chiari e completi.

POWER UTILITIES

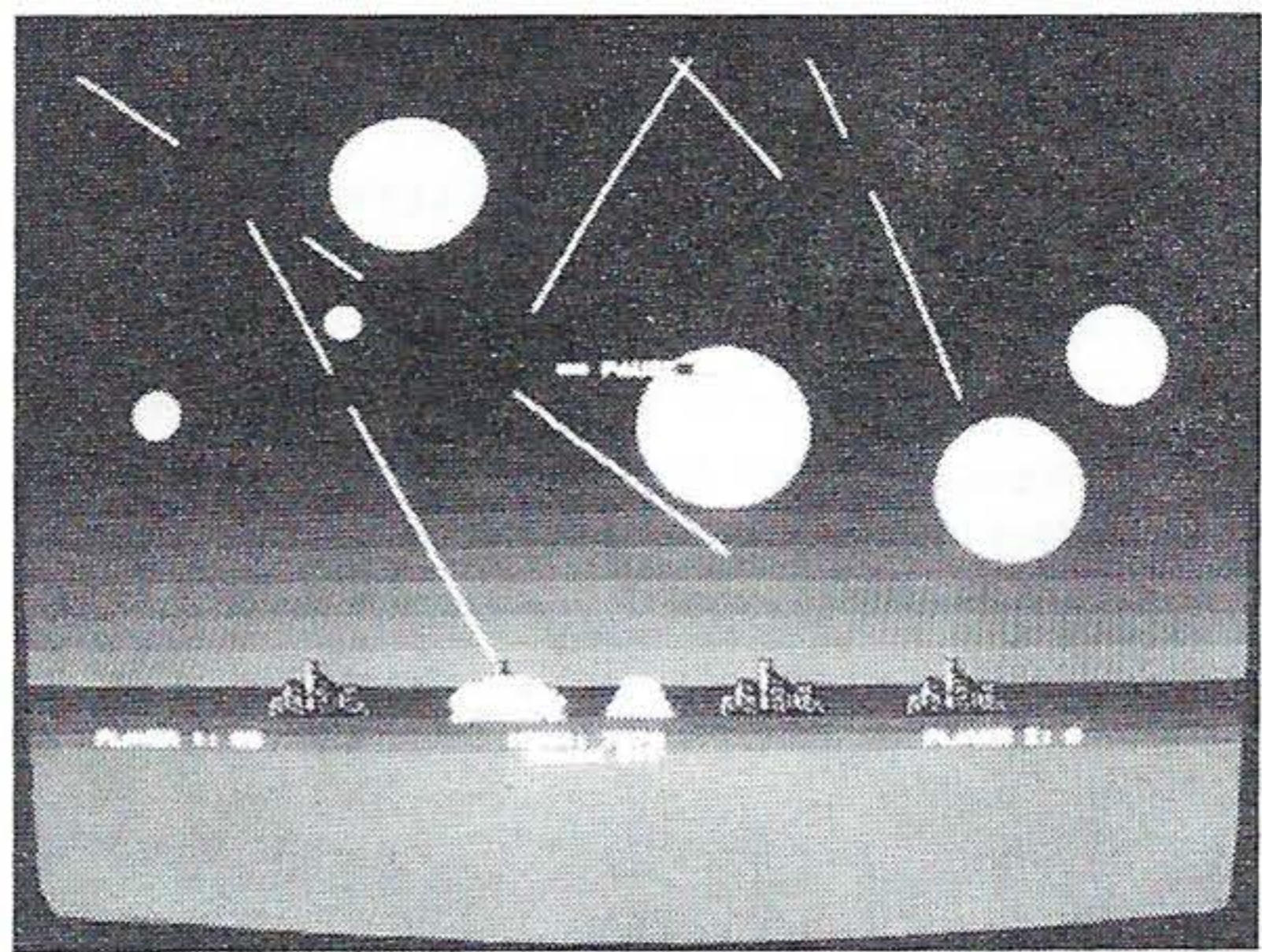
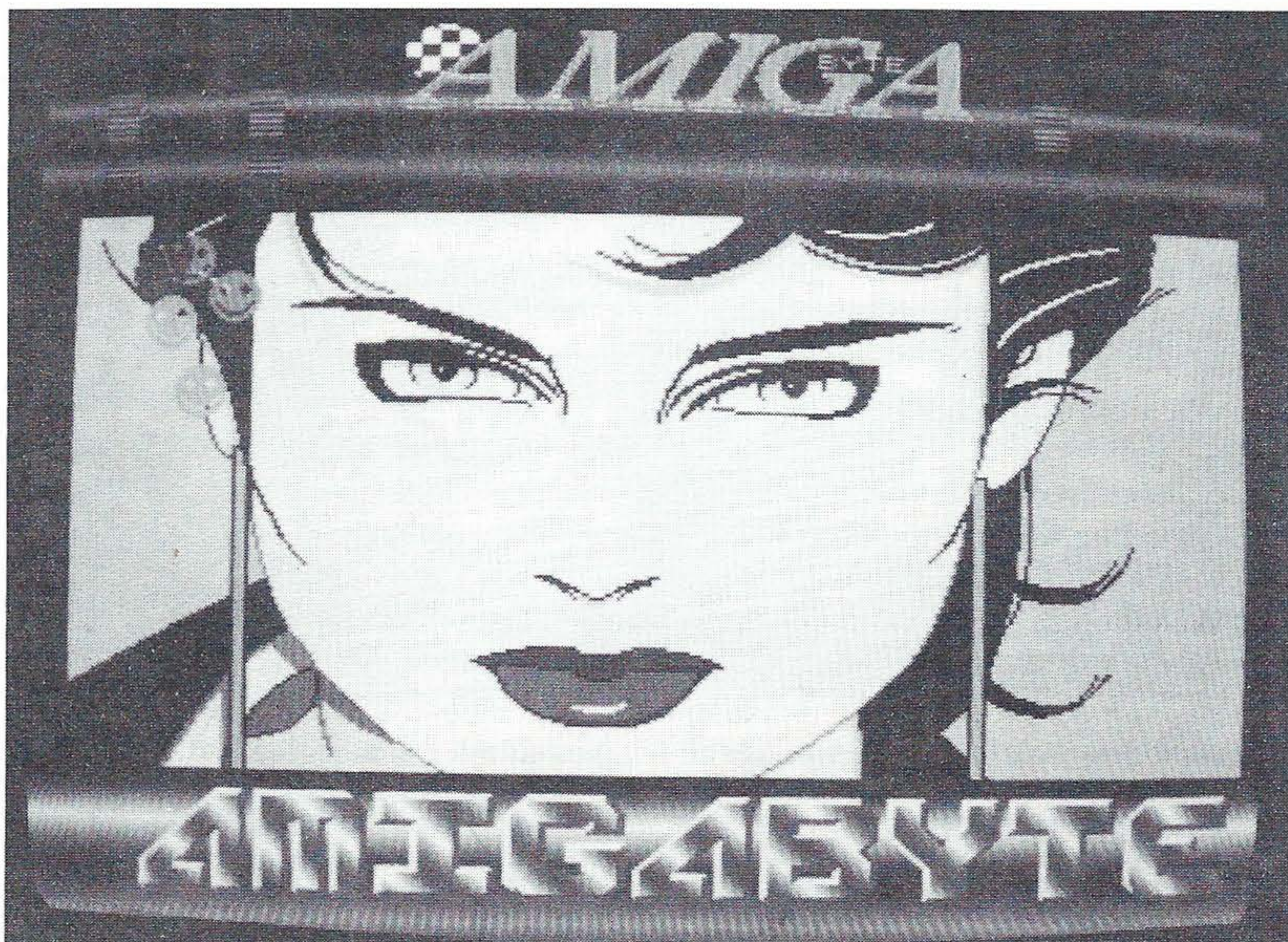
Oltre al materiale PD e shareware a cui abbiamo fino ad ora accennato, UGA distribuisce anche una serie di programmi originali a basso prezzo, venduti al pari di qualsiasi programma commerciale e perciò non liberamente distribuibili. Il più noto in questa serie, denominata Power Utilities, è l'ormai diffusissimo «PowerPacker Professional» (lire 29.000), giunto ormai alla versione 3.0B ed utilizzato da migliaia di utenti Amiga in tutto in mondo; AmigaByte ne ha già parlato diffusamente nella rubrica Tools del fascicolo 30, ed ha incluso la precedente versione 2.3B (shareware) nel dischetto allegato al fascicolo 18.
Altrettanto noto e valido è il programma «The Musical Enlightment» (lire 39.000), un software di composizione musicale basato sull'uso del pattern, simile come concezione ai numerosi derivati del SoundTracker. I risultati ottenibili con questa utility sono di qualità molto elevata, come testimonia il numero di demo in circolazione contenenti colonne sonore create con «The Musical Enlightment». In uno dei prossimi fascicoli ci occuperemo più in dettaglio di questo interessante tool, nell'ambito di una rassegna dedicata al miglior software musicale. Entrambi i programmi sono forniti con un assortimento di utility accessorie, e di sorgenti e routine indirizzate ai programmatori che volessero includere nei propri programmi brani musicali creati con «The Musical Enlightment», o caricare file compattati con «PowerPacker Pro». Anche questi programmi sono disponibili presso la redazione di AmigaByte, completi di documentazione tradotta in lingua italiana.

SHADOW MAKER

Proviene dal Canada una tra le utility certamente più interessanti per tutti coloro che operano nel settore del desktop video e della grafica con Amiga. «Shadow Maker» è un programma di conversione di font in grado, come rivela il nome, di aggiungere un'ombra a qualsiasi font standard Amiga e di trasformarlo in un font a colori (ColorFont) in un formato adatto per l'uso con qualsiasi programma di titolazione.
I font a colori sono diventati uno standard con l'introduzione della nuova release 2.0 del WorkBench; nelle versioni precedenti (fino alla 1.3) il loro utilizzo era vincolato al caricamento di un'utility residente, denominata «ColorText», che modificava alcune delle routine del sistema operativo e ne rendeva possibile il display con qualsiasi software grafico. «ColorText» viene originariamente fornito con «Deluxe Paint III» e con molti altri analoghi programmi di grafica e titolazione video, ed è indispensabile per poter caricare ed usare font a colori come quelli della celebre collezione «Zuma Fonts» o «Kara Fonts».
«Shadow Maker» trasforma qualsiasi font Amiga (per intenderci, quelli micromatici usati normalmente in ambiente Cli e dai programmi di trattamento testi) in un ColorFont a quattro colori: tre di essi sono usati rispettivamente per i caratteri, per lo sfondo e per l'ombra da proiettare su quest'ultimo.
Il quarto colore «Shadow Maker» lo utilizza (su richiesta) per generare l'antialiasing del font, ovvero per sfumare con una tinta intermedia i colori dei pixel adiacenti nelle linee diagonali e per ridurre quindi otticamente l'effetto a «scalini» tipico di alcuni font quando vengono ingranditi.
I risultati ottenibili con «Shadow Maker» sono eccellenti: invece di spendere tempo e denaro per procurarsi raccolte di font a colori da usare per titolazioni video o per generare scritte da impiegare nei propri demo, si possono convertire facilmente in font Amiga standard forniti con il WorkBench, aggiungendo loro un look più professionale grazie all'arrotondamento dovuto all'antialiasing ed alle eventuali ombre personalizzate.
È possibile definire il tipo di ombra (può essere proiettata sullo sfondo o estrusa, lasciando una scia), la sua lunghezza in pixel, la direzione dell'ipotetica sorgente di luce che la genera, e lo sfasamento in pixel rispetto al centro del carattere.
L'utility «On Top» fornita con il programma consente inoltre di evitare un inconveniente dovuto all'utilizzo di ombre troppo lunghe: una volta installata in memoria, essa provvede a controllare che, nelle parole, eventuali ombre dei singoli caratteri non debordino sovrapponendosi ai caratteri successivi, rendendoli illeggibili.
La versione demo liberamente distribuibile di «Shadow Maker» è perfettamente funzionante, ma comprende però la limitazione del salvataggio di font con dimensioni comprese tra 10 e 40 pixel.
Si tratta di un formato sufficiente al trattamento di moltissimi font per applicazioni generiche (testi in bassa e media risoluzione): la versione commerciale completa (il cui costo è di circa 20 dollari) del programma può essere richiesta all'autore e gestisce font fino ad una dimensione massima di 100 pixel.

Disponibile nel dischetto: Fish 460.

SUL DISCHETTO...

La Guru Meditation, croce e delizia di migliaia di programmatori Amiga, è tra le apparizioni più temute dalla maggior parte di chi usa il computer. La vista del famigerato rettangolo rosso lampeggiante sullo schermo è infatti quasi sempre sinonimo di condizione di errore,

blocco del sistema e perdita di dati. Non sempre le Guru o, più propriamente, i messaggi di Alert del sistema sono però accompagnati da un reset di Amiga: a volte essi possono essere generati ad arte da un'utility, come nel caso di ALERT, presente sul dischetto di AmigaByte di questo mese. Con questo piccolo tool chiunque potrà generare miniprogrammini che, una volta lanciati, visualizzeranno sullo schermo messaggi personalizzati lampeggianti in perfetto stile Guru Meditation, per richiamare l'attenzione di chi usa il computer o causargli un piccolo spavento.
Chi segue il quiz a premi televisivo «Bis» non avrà problemi a comprendere il meccanismo di AMIGATRATION, un gioco che mette a dura prova le capacità mnemoniche dei contendenti. I giocatori, da uno a quattro, devono cercare di accoppiare il maggior numero di caselle nascoste, scoprendone solo due alla volta e tenendo a mente la posizione dei simboli sottostanti.
TRIPPPIN (sì, è scritto proprio con tre «p»!) è una versione di un inedito gioco da tavolo americano nel quale due giocatori, uno dei quali può

Il dischetto allegato ad «Amiga-Byte» contiene i listati dei corsi e dei tutorial pubblicati sul fascicolo di AmigaByte ed alcuni programmi di pubblico dominio. Ogni cassetto contiene il programma, la sua documentazione originale in lingua inglese (quando risulta disponibile) ed una breve spiegazione in lingua italiana. Il software inviatoci dai lettori è generalmente accompagnato dalle istruzioni originali, con eventuali nostre aggiunte (se possono risultare utili all'utente).

Tutti i programmi presenti sul dischetto possono essere lanciati tramite il WorkBench o tramite il CLI. L'icona CLI presente sul dischetto di «Amiga Byte» serve unicamente per lanciare i programmi contenuti in questo dischetto: per ragioni di spazio infatti, il dischetto di «Amiga Byte» non contiene tutti i comandi AmigaDos normalmente presenti in un disco WorkBench standard.

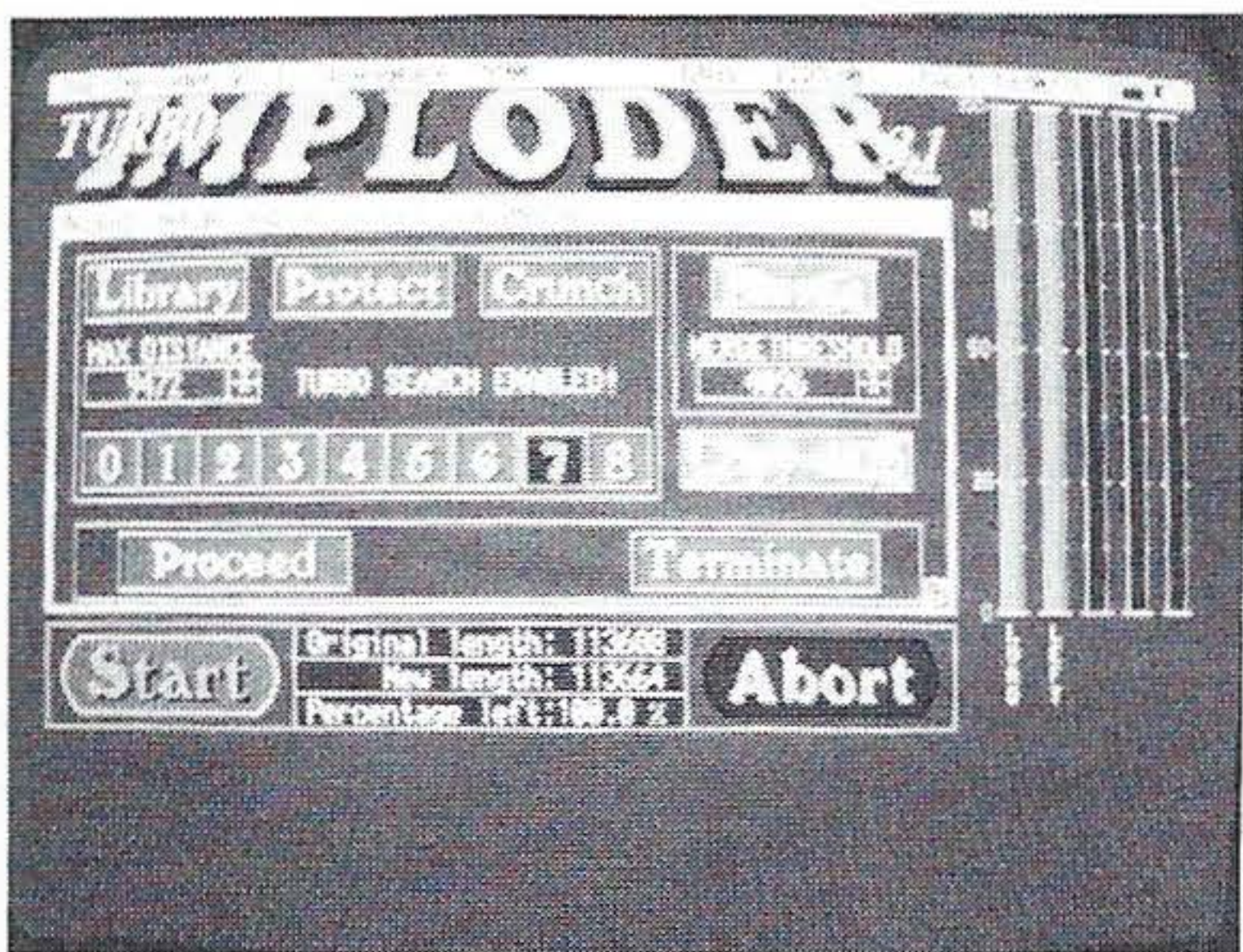
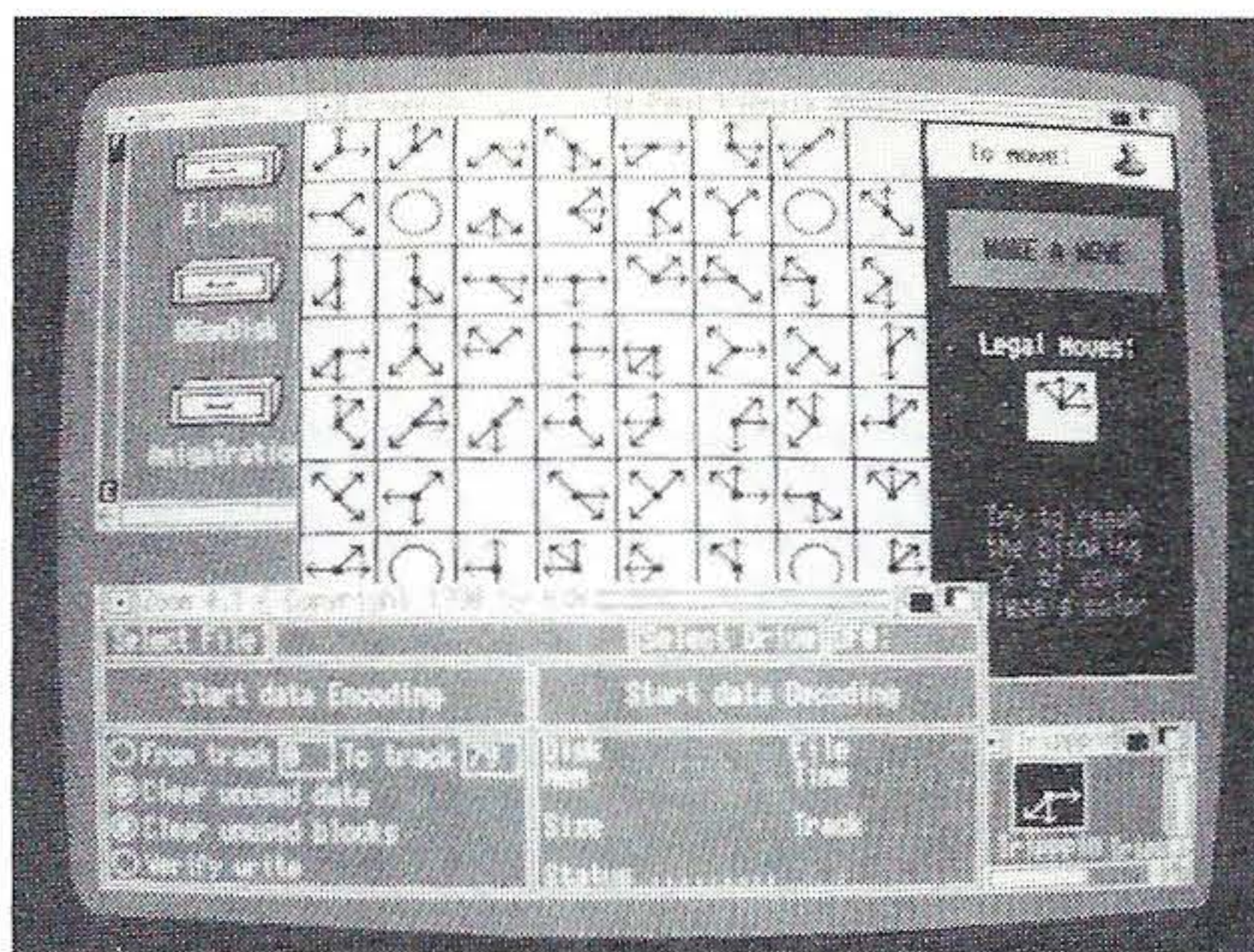
Per questa ragione il disco non contiene, ad esempio, i driver delle stampanti o il programma di configurazione «Preferences», il software di gestione della porta parallela o seriale, e le numerose altre utility del WorkBench, necessarie per la gestione di una stampante, di un modem o di altre particolari applicazioni.

Potete trasferire i comandi ed i file che vi servono (ad esempio il driver per il vostro modello di stampante) dal vostro disco WorkBench, dopo aver creato lo spazio necessario eliminando i programmi di questo disco che non vi interessano: ricordiamo però che è necessario agire su di una copia del dischetto di «Amiga Byte», in quanto un errore potrebbe portare ad una irreparabile perdita dei dati in esso contenuti.

I programmi vengono provati prima di essere distribuiti; tuttavia non ci risulta umanamente possibile garantire l'assoluta mancanza di eventuali piccoli «bachi».

Nel caso abbiate difficoltà con il funzionamento di un nostro programma, potete farcelo presente e tenteremo (per quanto ci è possibile) di aiutarvi; il nostro servizio di consulenza tecnica telefonica risponde ogni mercoledì pomeriggio presso la redazione di «Amiga Byte» dalle ore 15 alle 18, al numero 02-79.50.47. Considerate comunque il fatto che per il software incluso in questo dischetto avete pagato una cifra simbolica, tenuto conto del prezzo del supporto e del numero dei programmi, e che spesso anche i migliori prodotti commerciali pagati cifre considerevoli possono palesare malfunzionamenti.

«Amiga Byte» sostituisce qualsiasi dischetto non funzionante causa difetti di fabbricazione e/o duplicazione. È sufficiente rispedire il dischetto difettoso alla

anche essere il computer, tentano di giungere per primi ad una casella base situata su di un tabellone. L'unico problema sta nel fatto che la direzione di ogni movimento viene stabilita dal vostro avversario...
Una vera e propria pietra miliare nella storia dei videogames è MISSILE COMMAND, il cui tema è tornato tristemente di moda nelle cronache degli ultimi mesi, dominate da missili Scud e Patriot: riuscirà la contraerea da voi comandata a proteggere la città dall'attacco missilistico incombente?
A metà strada tra il gioco ed il programma educativo si colloca

AUTOMALAB, una variazione sul tema della popolare simulazione «Life»: tramite questo programma

potete riprodurre sul vostro computer un ipotetico ambiente di sviluppo di cellule, assistendo alla loro nascita ed influenzandone la crescita, per creare sullo schermo suggestive simmetrie «viventi».
Altrettanto interessanti dei giochi sono le utility del dischetto di questo mese: IMPLODER e ZOOM sono due programmi complementari, entrambi dedicati al tema della compattazione e dell'archiviazione dei dati. Il primo, versione shareware di un software precedentemente distribuito commercialmente, è un «cruncher» in grado di comprimere singoli file mantenendoli eseguibili; il secondo è un archiviatore di dischetti, che permette di trasformare un intero disco in un singolo file di dimensioni ridotte, che occupa men spazio e può essere ad esempio trasmesso più velocemente
e più economicamente via modem.
Completamente made in Italy è EL MASO, un programma che, a dispetto del nome, consente il tracciamento e lo studio di funzioni bidimensionali. Il suo impiego risulterà certamente utile agli studenti alle prese con esami di matematica e problemi di trigonometria.
A chi utilizza infine preferibilmente il Cli sono dedicati MOVESYS, che consente di ridefinire il volume e i dispositivi logici del disco di sistema con un solo comando evitando sfilze di noiosi comandi Assign, e RRAMDISK, un device driver che implementa un disco ram resistente al reset, dal quale è possibile effettuare il boot del computer. La marcia in più di questo Ram Disk, rispetto al normale disco RAD: disponibile con WorkBench 1.3, risiede nel fatto di essere dinamico, ovvero nell'occupare memoria solo quando al suo interno viene copiato qualcosa, e nel rilasciare la ram usata al sistema quando non più necessaria. □

---

redazione (l'indirizzo è indicato sulla rivista), allegando una lettera nella quale siano chiaramente specificate le seguenti informazioni in stampatello:
1) nome e cognome
2) indirizzo completo
3) NUMERO DEL FASCICOLO AL QUALE IL DISCHETTO ERA ALLEGATO

Prima di rispedire il dischetto, siate certi che i problemi non derivino da errori o da inesattezze nel caricamento o nell'esecuzione dei programmi: in particolare, leggete sempre le istruzioni allegate ai programmi per determinare se essi non richiedano particolari accorgimenti per il caricamento (espansioni di memoria, drive aggiuntivi, versioni particolari di KickStart o WorkBench, etc.).

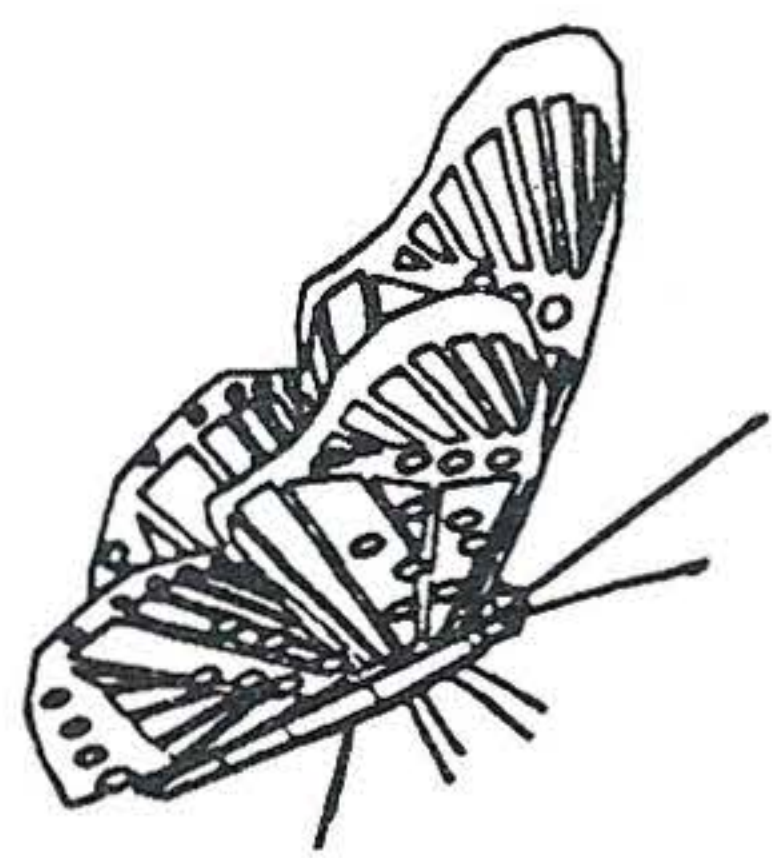
Gli errori di caricamento dovuti a difetti del supporto magnetico sono sempre segnalati da messaggi di errore del tipo «Read/Write error» o «Disk Corrupt» o «Not a Dos disk».

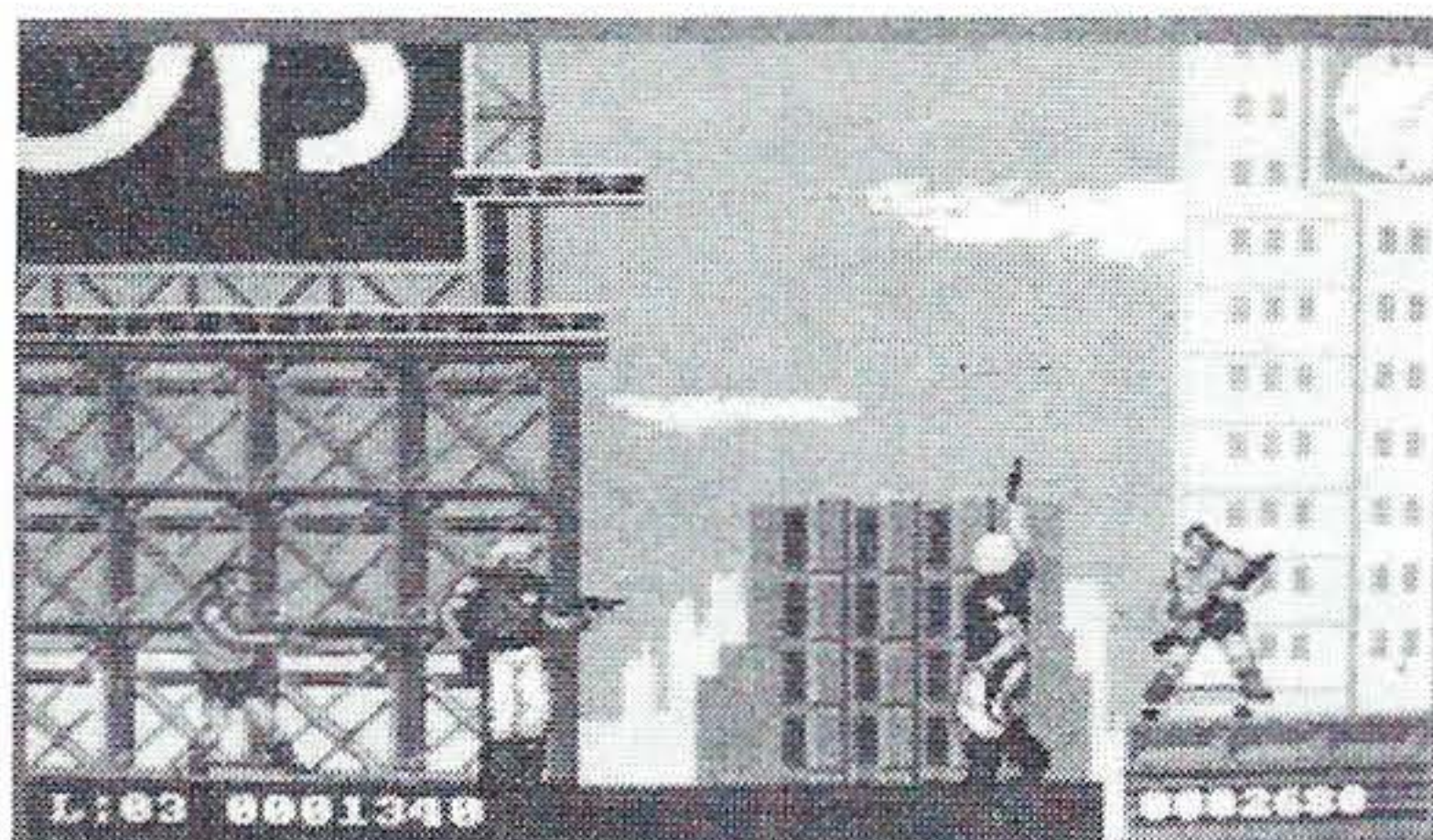
SE «NON FUNZIONA»

Ricordiamo che, a parte rarissime eccezioni, eventuali problemi di funzionamento e blocchi del sistema con la comparsa del messaggio di errore «SOFTWARE FAILURE - GURU MEDITATION» sono relativi a problemi di carattere software e NON a difetti del dischetto. In questi casi pertanto la sostituzione del dischetto da parte nostra è inutile e non risolve il problema, che probabilmente è invece dovuto a qualche conflitto con programmi residenti in memoria o a scarsità di Ram.

In questi frangenti, ed in particolar modo quando si è in possesso di Amiga privi di espansione di memoria, può essere utile attenersi alla seguente procedura, che consente il massimo risparmio di memoria possibile ed assicura che la memoria di Amiga sia libera da programmi che potrebbero creare problemi di compatibilità:
— Accendete il computer ed inserite il dischetto di «AmigaByte» nel drive interno.
— Non appena appare lo schermo con la finestra Cli, premete i tasti CTRL-D e teneteli premuti fino a quando non appare il prompt di AmigaDos (ovvero «1>») ed il cursore.
— Digitate il comando «ADD21K» e premete Return. Questo comando, presente nella directory C dei dischetti di «Amiga Byte», libera circa 20K di memoria Chip togliendo due colori dallo schermo WorkBench (il cursore cambierà infatti colore).
— Digitate il comando «CD» seguito da uno spazio, da due punti, e dal nome del cassetto in cui è contenuto il programma da caricare. Esempio: se volete caricare il programma «Pippo» presente nel cassetto «Pluto», digitate «CD :PLUTO» e premete Return.
— Digitate il nome del programma da caricare, eventualmente seguito dai parametri necessari al suo corretto funzionamento. Esempio: per caricare il programma «Pippo», digitate «PIPPO» e premete Return.

Esempio: per caricare un programma chiamato «LhWarp», la cui icona si trovasse nel cassetto «Utility», occorrerebbe usare questi comandi:
1> CD :UTILITY
1> LHWARP
(N.B.: i caratteri «1>» rappresentano il prompt di Amigados che appare prima del cursore, e non devono essere digitati dall'utente).

Consigliamo comunque di leggere MOLTO ATTENTAMENTE le istruzioni di ogni programma prima di eseguirlo, in quanto alcuni programmi possono richiedere particolari accorgimenti o requisiti hardware/software per il caricamento e/o l'esecuzione.

SUGGERIMENTI E TRUCCHI VARI

Recensito nel fascicolo 31 di AmigaByte, «E-Swat» è un gioco d'azione della US Gold che ricorda vagamente «Robocop». Per attivare il cheat mode, basta mettere il gioco in pausa e digitare la frase «JUSTIFIED ANCIENTS OF MU MU» (provate con o senza spazi). Se il codice è

stato digitato correttamente, il bordo dello schermo lampeggerà brevemente e disporrete di 99 vite extra.

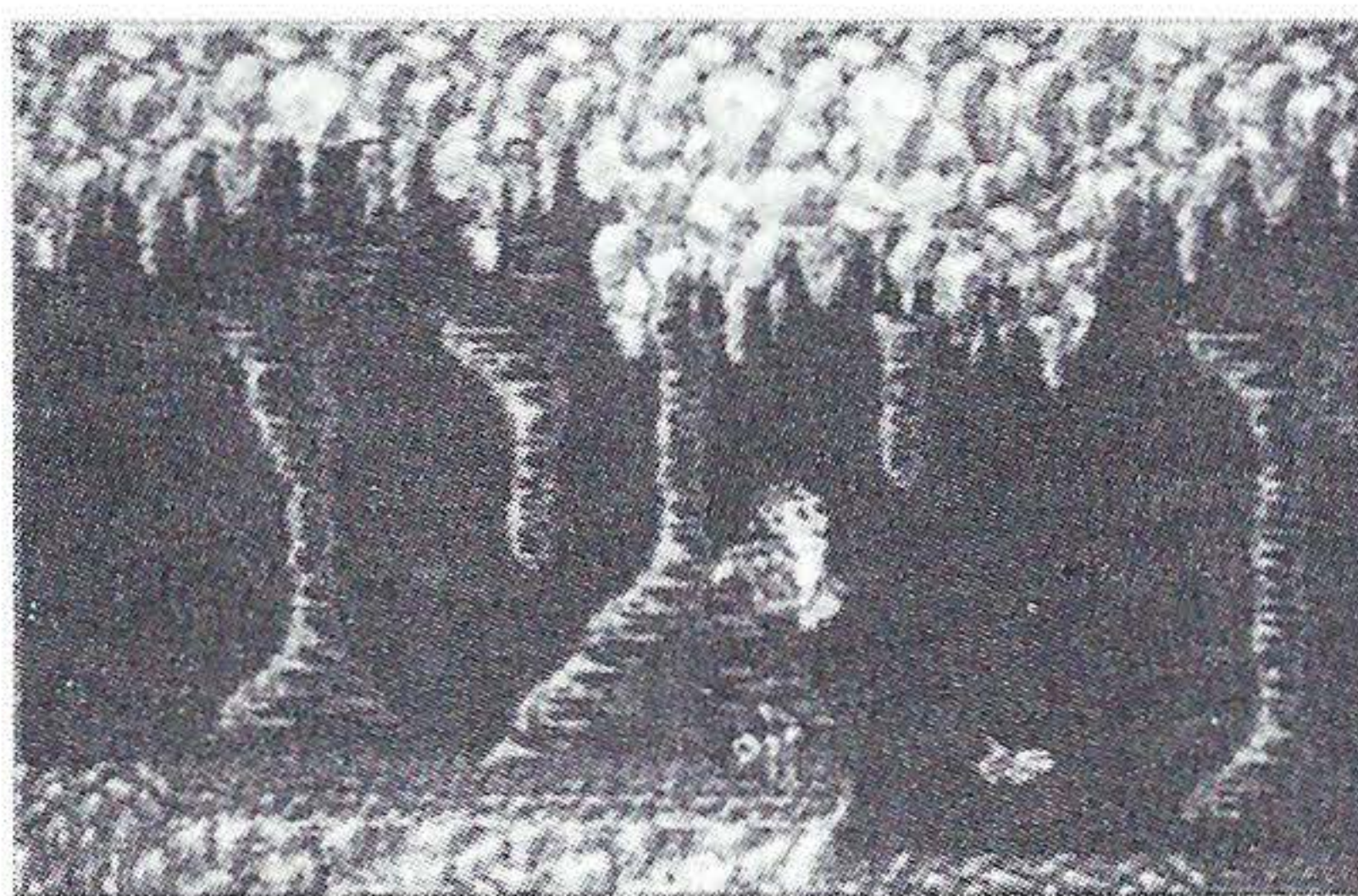
Selezionate il livello di gioco nello shoot'em up «Wings of Fury» della Domark e poi digitate la frase «COLIN WAS HERE» (compresi gli spazi). A questo punto avrete attivato il cheat mode e potrete selezionare qualche funzione speciale mediante alcuni tasti:
C = cambia le armi a disposizione
M = armi infinite
P = aggiunge una vita extra
D = invulnerabilità
F = rifornimento carburante

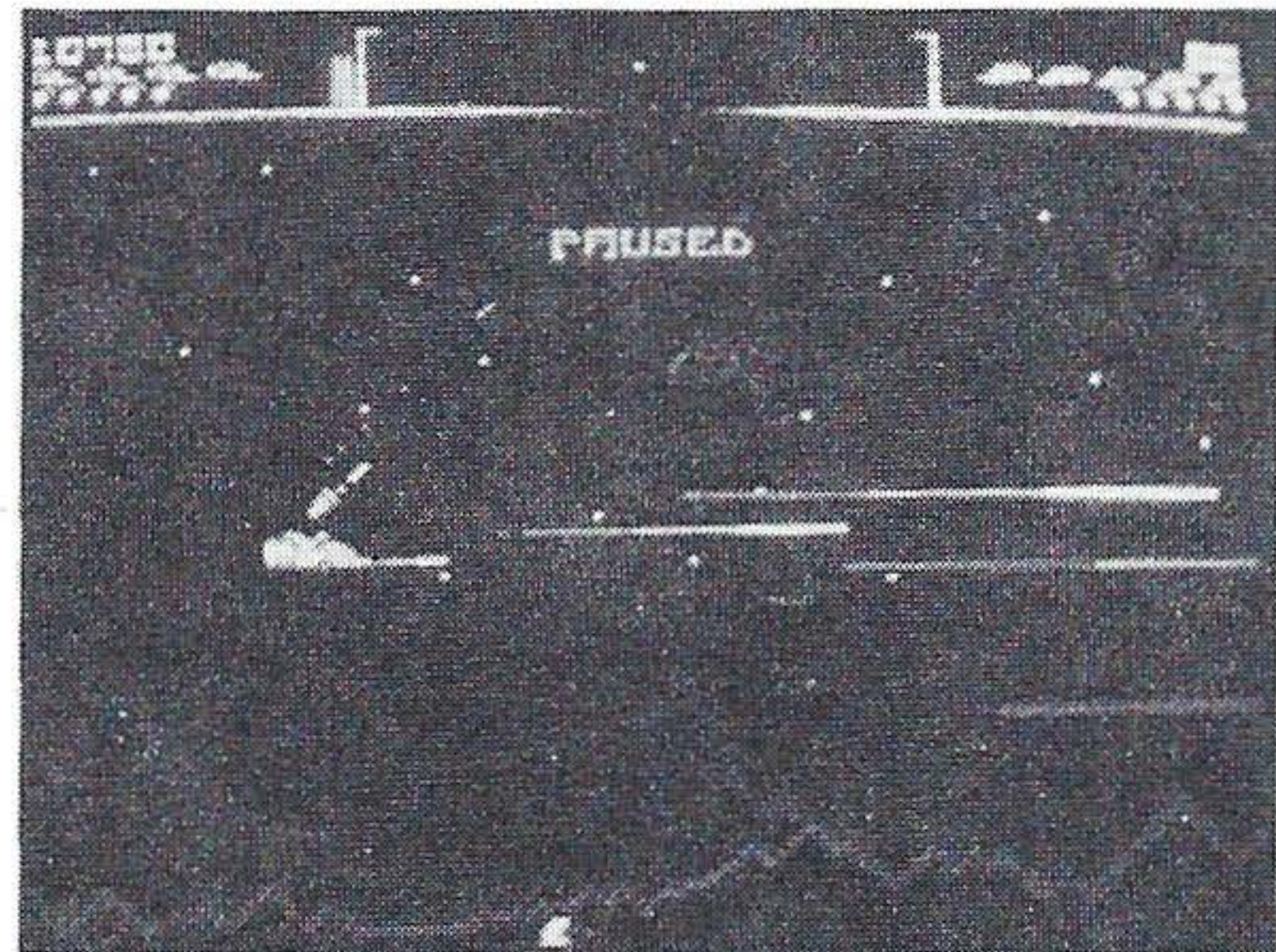
Come nei precedenti «Space Ace» e «Dragon's Lair», la ReadySoft ha inserito anche in «Dragons' Lair II: Time-Warp» un cheat mode per poter portare a termine l'intera avventura senza rischi: è sufficiente, nella schermata dei titoli, premere Return e poi digitare la frase «GET MORDROC DIRK» (spazi compresi).

Il gioco a basso prezzo «Treasure Island Dizzy» della CodeMasters è stato tra i più venduti della scorsa stagione in Gran Bretagna. Per emulare le imprese di Icaro, potete digitare la parola ICANFLY all'inizio del gioco: tirando il joystick verso l'alto scoprirete di essere in grado di volare sopra gli ostacoli.

Anche il simpatico arcade adventure «Horror Zombies From the Crypt» della Millennium può essere reso ancora più facile: basta premere contemporaneamente i tasti «M», «,» e «.» per poter disporre di vite infinite. Digitando inoltre la parola CHEATMODE durante il gioco potrete premere F10 per saltare al livello seguente.

Nella schermata introduttiva di «Enchanted Land» della Thalion, digitate le parole «TCB RULES FOREVER». Lo schermo lampeggerà e potrete premere F3 per accedere ad un editor di opzioni. Premendo inoltre F2 e la barra spaziatrice du-

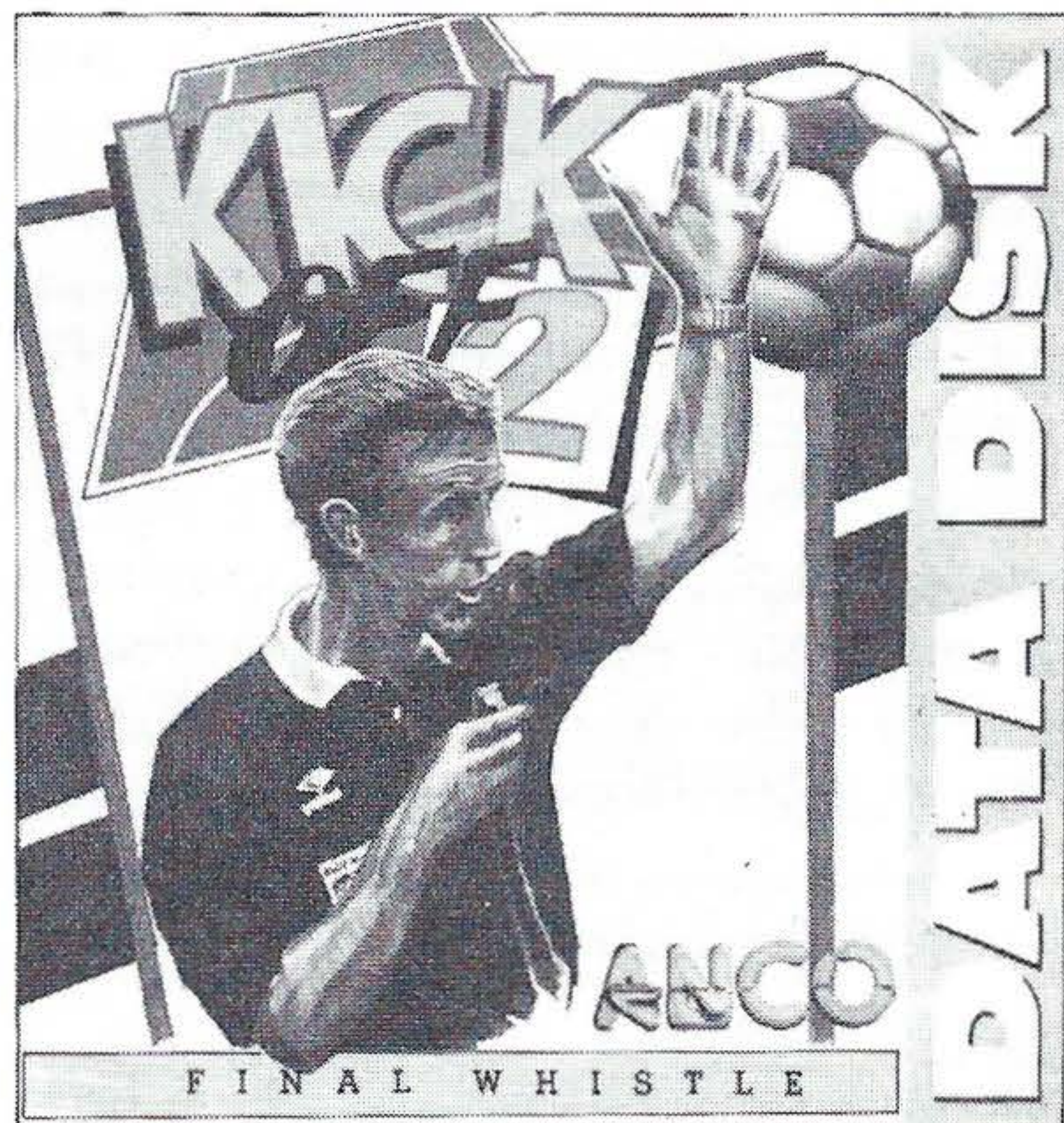
rante il gioco, verrete teletrasportati per fronteggiare il guardiano di fine livello.

Un tuffo nel passato con il platform game «Manic Miner 2»: attivate la pausa con la barra spaziatrice, poi premete «*» e togliete la pausa per aggiungere 9 vite extra al vostro intrepido minatore.

Il classico shoot'em up spaziale Defender sta attraversando recentemente un periodo di rinnovata popolarità grazie a «Defender II», una nuova versione di Jeff Minter, un tempo noto come programmatore di alcuni celebri giochi per il Commodore

64 come «Attack of the Mutant Camels».

Durante il gioco, per attivare il cheat mode digitate la parola «GOATY». Per accedere ai vari livelli di gioco, usate invece questi codici:
LEMAC
ZIPPO
LAZER
DAFAD
MAMOG
FUNKY
DONKY

Dopo aver iniziato a giocare a «Strider II» della US Gold, premete contemporaneamente il tasto F2 e tutte le lettere che compongono la parola STRIDER. In un angolo dello schermo apparirà una faccia sorridente e potrete premere «E» per aumentare la vostra energia, «D» per aumentare quella del vostro droide e «T» per aumentare la velocità. Con il tasto Help si esce invece dal cheat mode.

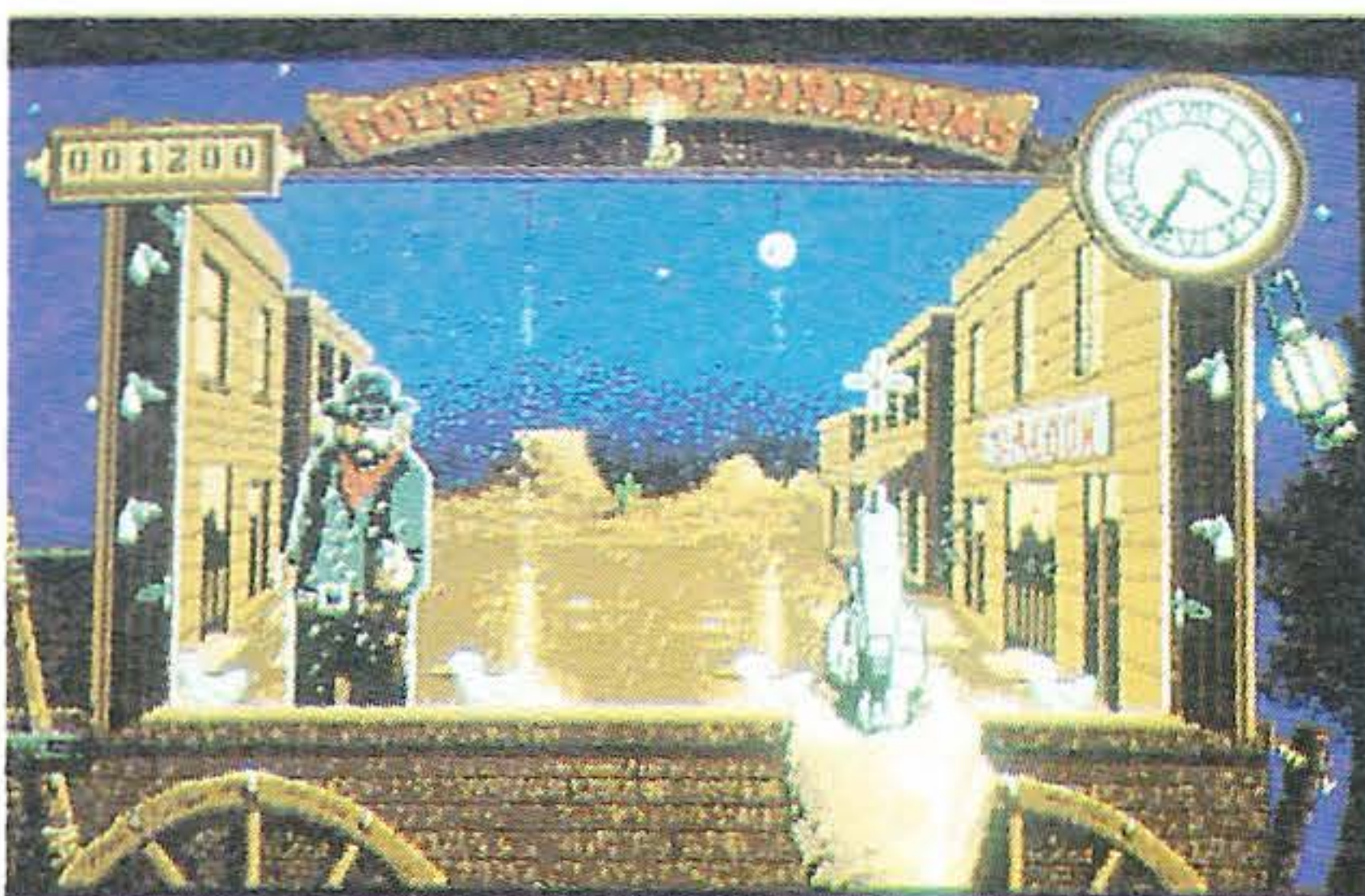
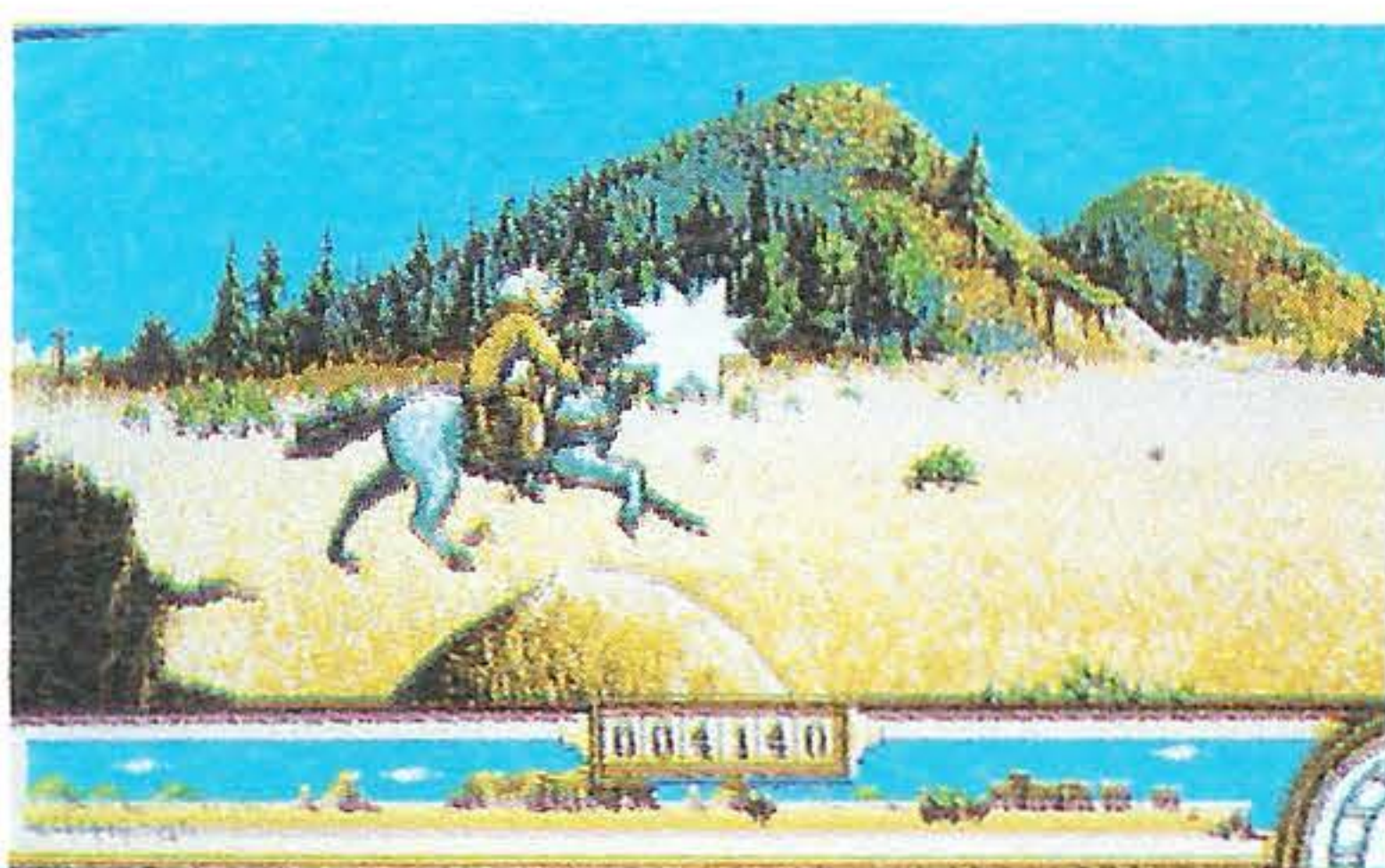
Nel meritatamente popolarissimo «Kick Off 2» della Anco, tenete premuto il tasto della lettera «R» quando state per subire un rigore e potrete vedere in anticipo la direzione nella quale il computer sta per calciare.

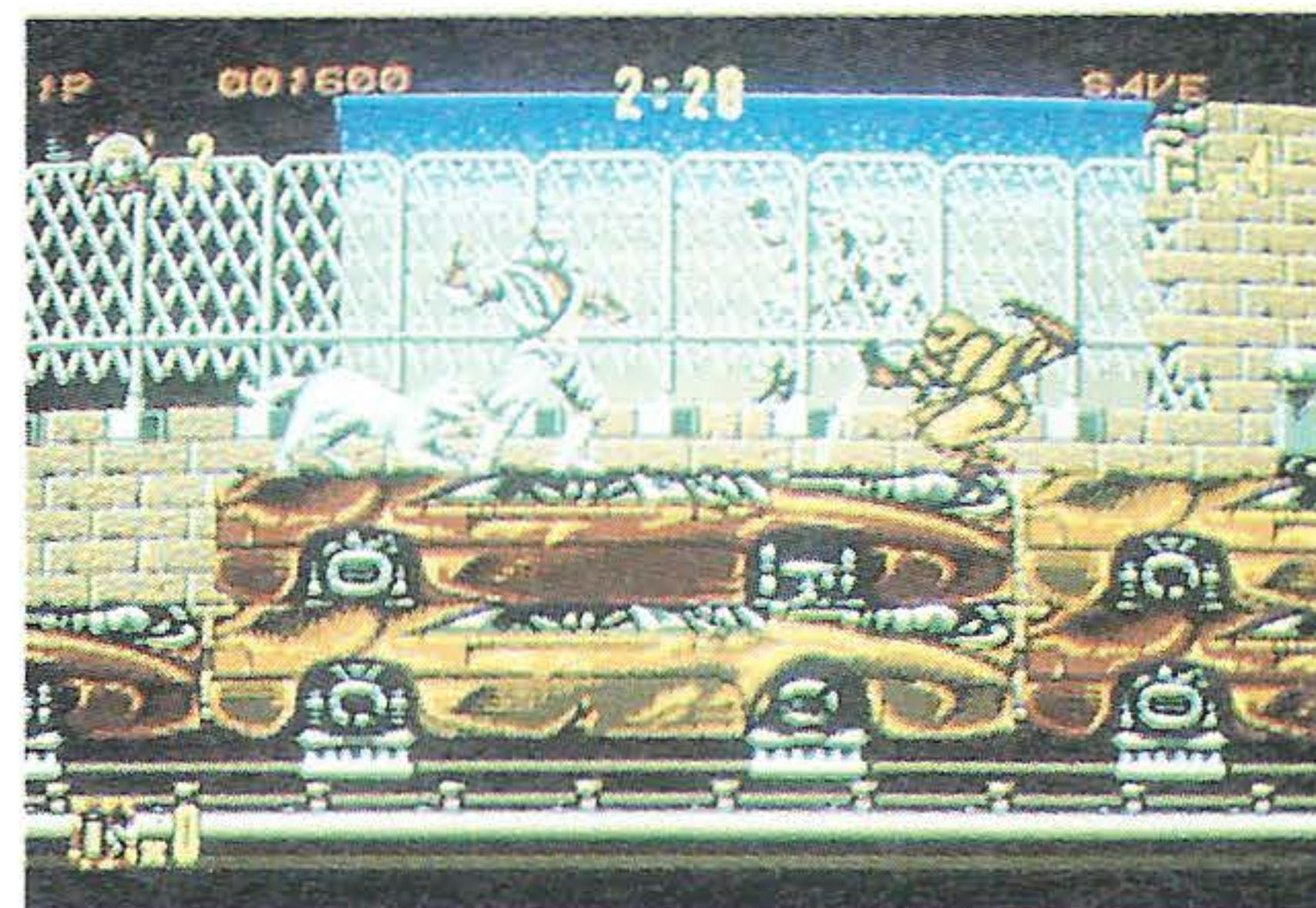
Giuseppe Lugano da Camerino ci segnala inoltre che, per avere più possibilità di vincere il campionato, conviene selezionare per ogni squadra i giocatori più scadenti (ad esempio un attaccante in porta, un difensore in attacco, etc.).

GEM X

Dopo tanti giochi ispirati o convertiti direttamente da arcade di origine giapponese, eccone finalmente uno programmato interamente nella terra del sol levante.
«Gem X», della software house nipponica Kaiko, è un puzzle-arcade originale, impegnativo e terribilmente frustrante che non sfigura affatto nel confronto con alcuni classici del genere come «Plotting» o «Puzznic».
Lo schermo di «Gem X» è suddiviso verticalmente in due sezioni, ciascuna delle quali contiene un gruppo di gemme.
Voi agite nella parte di sinistra e dovete riuscire a sistemare le gemme in modo che

siano disposte esattamente come quelle della parte destra.
Tra le due porzioni di schermo sono disposte alcune gemme che ricordano la sequenza di colori (rosso, verde, blu, rosa e giallo) su cui è basato il meccanismo del gioco.
Selezionando una gemma, essa assume il colore di quella presente nello schema due posizioni dopo (ad esempio una gemma verde diventa rosa), mentre quelle immediatamente circostanti cambiano colore in quello successivo (ad es. una gemma rossa diventa verde). Le gemme gialle, essendo le ultime nello schema, non cambiano colore e scompaiono semplicemente, facendo precipitare le altre e contribuendo ulteriormente ad incasinare la situazione.
Gli schemi da completare sono 26, ma già il primo è un osso duro. Fortunatamente al termine di ogni livello appare una fanciulla discinta che comunica il codice necessario per ripartire da quel punto in eventuali partite future.
La spettacolarità della grafica non è mai un punto di forza di questo genere di gioco, ed anche «Gem X» non fa eccezione: ma il sonoro, l'originalità e la giocabilità ripagano ampiamente dell'assenza di animazioni ed effetti speciali. Un debutto davvero promettente per la Kaiko: se i giapponesi si dedicheranno al software Amiga con la stessa determinazione e l'abilità dimostrata in altri settori della tecnologia, ne vedremo davvero di tutti i colori.

BACK TO THE FUTURE III

I precedenti due episodi della saga cinematografica di «Ritorno al futuro» avevano generato altrettanti giochi arcade per Amiga senza infamia e senza lode. Anche la terza ed ultima puntata è stata ora puntualmente convertita a 16 bit ed il risultato, per quanto non memorabile, appare leggermente superiore ai predecessori.
Basato su quattro sequenze-chiave del film, «Back to the Future III» è ambientato tra cavalli e cowboy nel West del 1855. Nella prima parte Doc deve salvare una donna a bordo di una diligenza guidata da cavalli imbizzarriti e diretta verso un precipizio, raggiungendola prima che avvenga il peggio.
La seconda sequenza è invece ambientata in un carrozzone da circo: dovete cimentarvi con il tirassegno e colpire abbastanza bersagli per guadagnarvi una fiammante Colt.
La terza parte vede Marty alle prese con una gang di cowboy malintenzionati, con un mucchio di piatti di torta come unica arma a disposizione. Usandoli come frisbee, dovete colpire ed abbattere tutti i nemici mentre

escono dal saloon, prima che loro possano ridurvi a groviera a colpi di revolver.
Nella quarta sequenza, infine, avete l'occasione per tornare al presente con la macchina del tempo. Per saltare a bordo dell'automobile De Lorean, dovete raggiungere la locomotiva di un treno in corsa che sta spingendola per farle guadagnare la velocità necessaria a tornare al futuro.
Grafica e sonoro sono di buon livello, anche se i quattro livelli non sono particolarmente impegnativi, e tutto il gioco è permeato da un'aria di *deja vu* che non garantisce certo a «Back to the future» una grande longevità.

SHADOW DANCER

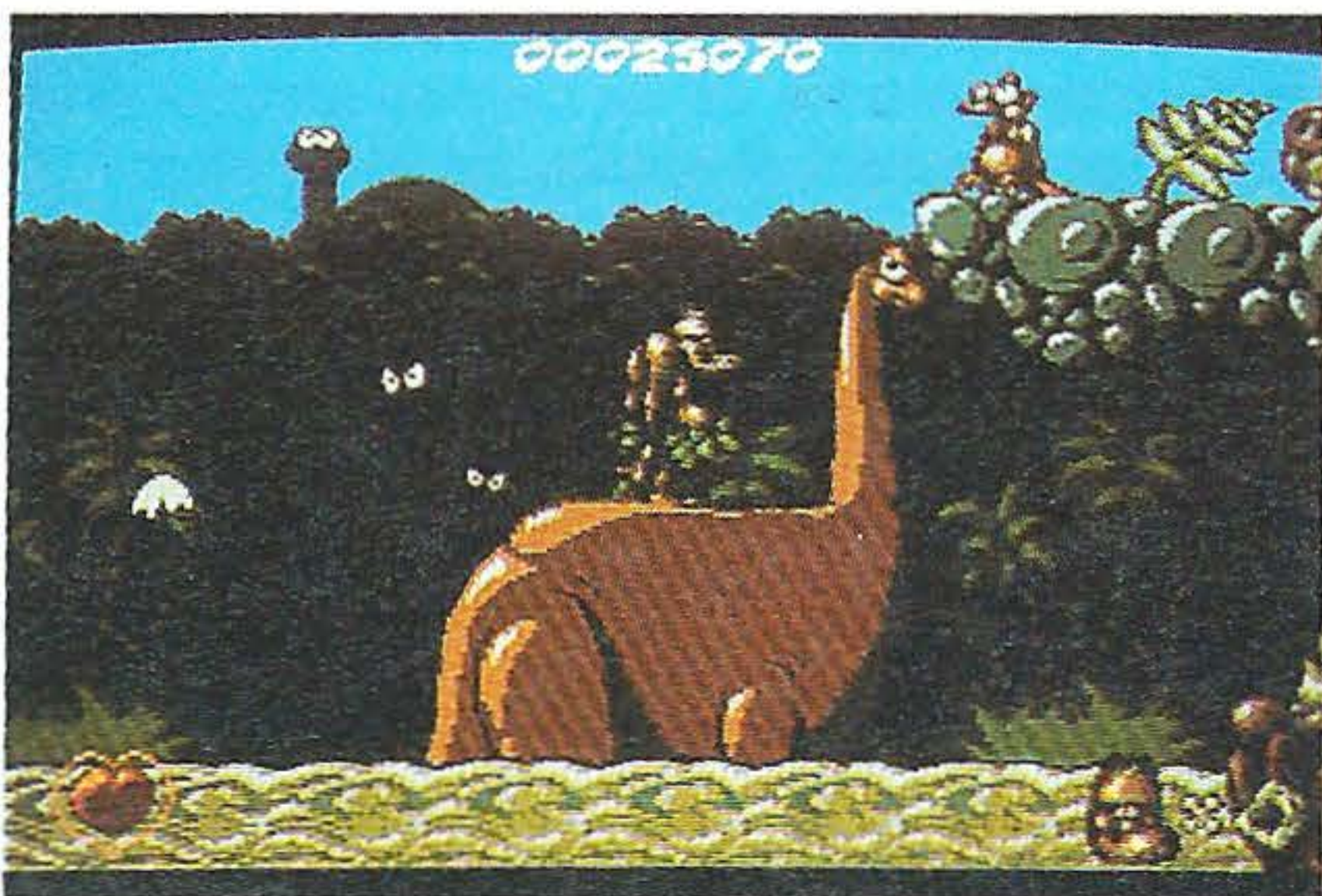
Seguito del popolare «Shinobi», questo «Shadow Dancer» è un ennesimo gioco di combattimento con arti marziali a scrolling orizzontale, non particolarmente originale o spettacolare.
Il giocatore impersona il solito ninja, dotato del tradizionale assortimento di armi (spada

e shuriken, ovvero le stesse appuntite da lanciare). In più, il protagonista è accompagnato da un cane, che lo segue fedelmente nei suoi spostamenti e che può essere aizzato contro i nemici tirando il joystick verso il basso e premendo il pulsante.
Il ninja è dotato di poteri magici e può, premendo lo spazio, invocare un incantesimo con il quale spezzare via ogni nemico presente sullo schermo. Gli incantesimi disponibili, di potenza crescente, sono quattro ed operano alla stessa maniera di quelli di «Golden Axe».
Scopo del gioco è percorrere tutti i livelli alla ricerca di una serie di bombe da disinnescare, ognuna delle quali è ovviamente protetta da un nutrito gruppo di loschi figuri armati di spade, fucili ed altre diavolerie.
Oltre ad essere poco originale, «Shadow Dancer» non è nemmeno particolarmente ben fatto. I movimenti degli sprite non brillano certo per il loro realismo, la grafica non è tra le più spettacolari viste

Software Express

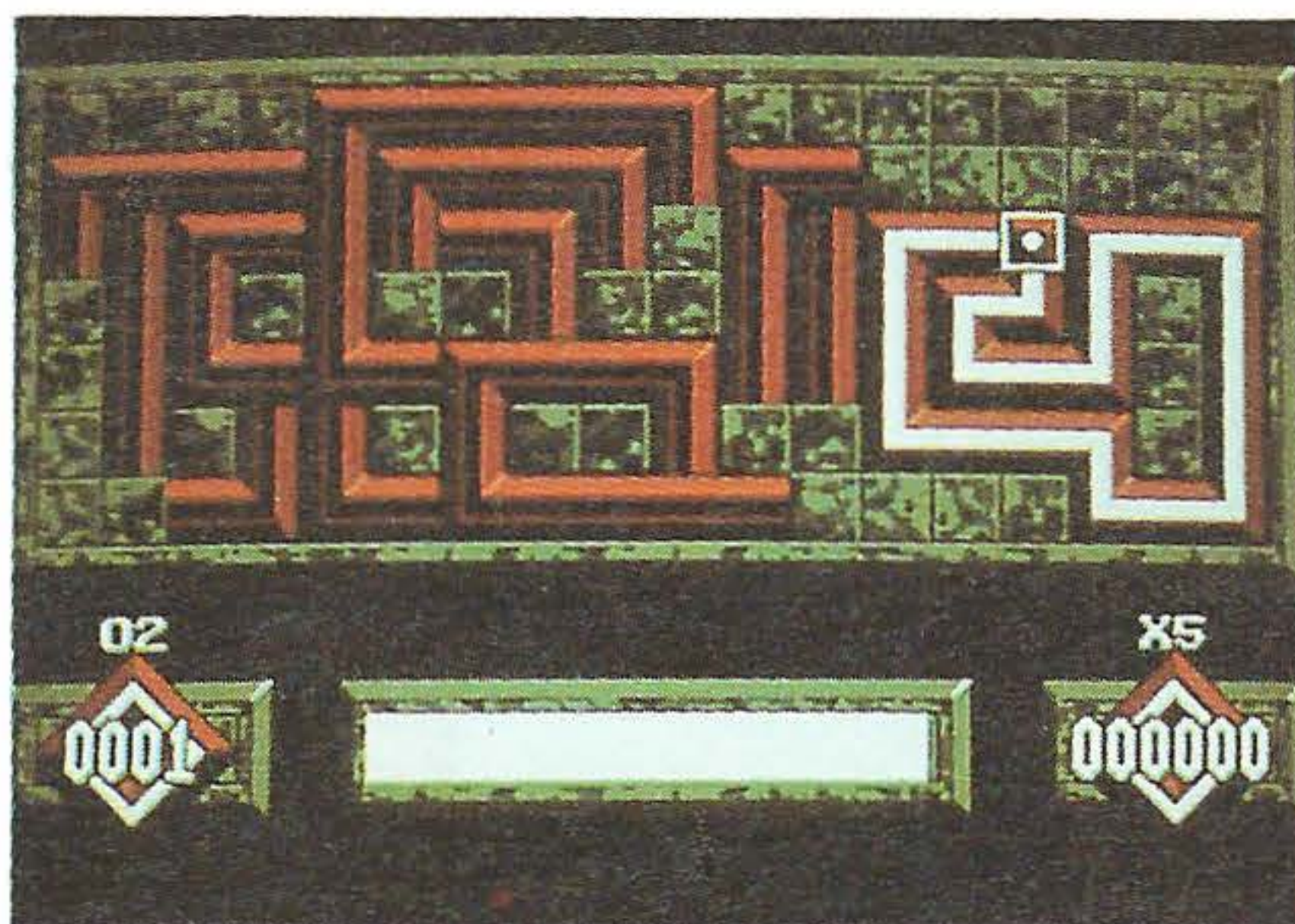
ultimamente su Amiga ed anche la giocabilità si mantiene su livelli mediocri. «Shadow Dancer», prodotto dalla US Gold, non è insomma un capolavoro: piacerà quasi certamente agli appassionati dei giochi di combattimento (i cosiddetti *punch'em up*) ed agli estimatori di «Shinobi», ma resta un gioco di livello mediocre, apparentemente sfornato in tutta fretta per approfittare della relativa popolarità del primo episodio.

CHUCK ROCK

Nella spettacolare introduzione animata, che da sola quasi vale il prezzo del programma, di questo gioco ambientato nella preistoria il malvagio Gary Gritter ha rapito la povera Ophelia mentre era intenta a stendere i panni in cortile, ed è compito di Chuck Rock

strapparla dalle grinfie del suo catturatore. Chuck deve attraversare cinque livelli (alcuni dei quali sotterranei o subacquei), popolati da dinosauri, pterodattili ed altre strane creature, per raggiungere la sua amata. Per evitare i pericoli ed eliminare i nemici, Chuck può saltare, gettare macigni o abbattere gli avversari con colpi del suo ventre prominente (!). Le rocce, in particolare, sono indispensabili per poter superare gli ostacoli: Chuck può sollevare pietre e spostarle opportunamente per usarle come piattaforme per saltare, oppure farsene scudo per proteggersi dalla caduta di macigni.
Il tono del gioco è decisamente umoristico e ricorda molto i cartoni animati della serie «Gli Antenati»: Chuck può farsi trasportare in groppa da brontosauri per superare le paludi ed in volo da pterodattili amici, e può usare coccodrilli come catapulta per raggiungere piattaforme troppo alte.
Ogni livello termina con un super-nemico da eliminare (un triceratopo, un mostro marino, una tigre), ma anche le insidie presenti nei vari schermi non sono da sottovalutare: anche i dinosauri fanno i loro bisognini di

tanto in tanto, quindi attenti quando gli passate tra le gambe, perché potreste ritrovarvi coperti non proprio di gloria...
«Chuck Rock» è un platform game di qualità superba: le animazioni degli sprite (sia del troglodita protagonista che dei numerosi animali preistorici) sono divertentissime e realistiche, ed i livelli sufficientemente vari ed impegnativi, pur senza essere impossibili da portare a termine.
Gli sfondi ed il sonoro sono allo stesso elevato livello, contribuendo a fare di questo gioco della Core Design uno tra gli arcade più brillanti degli ultimi mesi.

LOOPZ

Oltre alle stragi di alieni ed ai raid aerei oltre le linee nemiche, i computer offrono anche passatempi meno cruenti ma non per questo meno estenuanti. Chiunque abbia raggiunto i livelli più avanzati di «Tetris» o «Klax» e sia sopravvissuto per raccontarlo sa perfettamente cosa intendiamo.
«Loopz» appartiene alla stessa categoria di rompicapo arcade, ed è parente stretto di «PipeMania» della LucasFilm: anche qui lo scopo è unire segmenti di «tubo» tra loro, con la differenza che per completare un livello non occorre tracciare percorsi lunghi e complessi. È sufficiente che i due capi della tubatura si uniscano, formando un poligono chiuso, cioè un «loop». I pezzi appaiono uno alla volta e, tramite il mouse, possono essere posizionati ovunque sullo schermo e ruotati di 90 gradi alla volta con il tasto.
Una volta chiuso un loop, il livello è completato ed il loop scompare: ma i pezzi avanzati presenti nel piano di gioco non svaniscono e continuano ad occupare spazio. Perciò gran parte della strategia risiede nel giocare d'anticipo e nel collocare oculatamente sullo schermo anche i pezzi che al momento non servono, in previsione di doversene sbarazzare in futuro. Come se

non bastasse, con l'avanzare dei livelli si riduce sempre più il tempo a disposizione per pensare e collocare ogni pezzo.
«Loopz» ha un meccanismo di gioco semplicissimo da imparare ma non per questo risulta poco impegnativo: al contrario, già dopo i primi cinque o sei livelli, la difficoltà comincia ad essere tale da scoraggiare anche chi è dotato di riflessi sovraumani.
Sebbene non appaia destinato a lasciare un'impronta duratura nella storia dei «puzzle» computerizzati, «Loopz» è un ottimo passatempo, intelligente ed anche abbastanza frenetico da piacere anche agli amanti dei giochi d'azione.

MOONSHINE RACERS

Ike e suo cugino Billy Joe sono due *moonshiners* ovvero, secondo lo slang americano, due distillatori casalinghi di whisky illegale. Ike produce l'intruglio e Billy Joe ha il compito di consegnarlo alla taverna di Tucker con il suo scassatissimo camion, cercando di arrivare prima dell'agguerrita concorrenza e soprattutto di evitare l'auto dello sceriffo Fat Sam.

L'originalità di «Moonshine Racers», della Millennium, risiede nel fatto che, una volta tanto, il giocatore è alle prese con una simulazione di corsa automobilistica nella quale non siede dietro al volante di una fiammante Ferrari o di una potente vettura da corsa, bensì di una carretta traballante che fatica a superare gli avversari e che rischia in ogni istante di fondere il motore. Un termometro nella parte superiore sinistra dello schermo consente di tener d'occhio la temperatura dell'acqua nel radiatore e di rallentare all'occorrenza per evitare il peggio; a destra invece appaiono le comunicazioni radio scambiate tra i vari personaggi del gioco, intercettate dal CB montato sul vostro camion.
Per migliorare le prestazioni del mezzo di trasporto potete acquistare accessori e pezzi di ricambio, da installare nel vostro garage. In casi estremi si può ricorrere al trucco di versare un po' di moonshine nel serbatoio, premendo la barra spaziatrice, che funge da «effetto turbo»: ma attenzione a non usarla tutta, o non ve ne resterà più da consegnare a Tucker...
«Moonshine Racers» è divertente e relativamente originale, ma sotto sotto altro non è che un gioco di corsa automobilistica nemmeno poi tanto spettacolare e realistico. Grafica e giocabilità sono discrete, ma una volta cessato il divertimento legato alla novità della trama, il gioco diventa monotono. Una nota di plauso, invece, per la simpatica musica stile country and western.

SUPERCARS 2

Quando ancora i computer non avevano raggiunto l'attuale diffusione, uno tra i giochi più popolari era quello delle automobiline: chiunque sia cresciuto in un'epoca in cui tra coetanei non si discuteva di chip ram e di risoluzione grafica avrà certo posseduto più d'un modellino di automobile e una pista sul quale farle gareggiare contro quella degli amici.
«SuperCars 2» rievoca quel divertimento, mettendovi al comando di una minuscola macchinina su un percorso ripreso dall'alto; e, tanto per rammentarci che siamo negli anni novanta, introduce qualche piccola innovazione tra gli optionals montati sulla vettura (cannoni, mine, mitragliatori ecc.). Prodotto dalla Gremlin Graphics,

«SuperCars 2» è il seguito di un popolare gioco di corsa uscito circa un anno fa, intitolato (chi lo avrebbe mai detto?) «SuperCars».
Il meccanismo di gioco, simile a quello dell'arcade «Super Sprint», è semplice: occorre giungere al termine di ogni percorso classificandosi entro i primi cinque (su un totale di dieci auto in gara) per poter passare al successivo. Ed ogni mezzo violento è lecito, anzi consigliabile, per raggiungere questo obiettivo. Ogni auto può sopportare una certa quantità di danneggiamenti, superata la quale è buona per lo sfasciacarrozze.
I percorsi, tortuosi come solo la fantasia malata di un programmatore può concepirli, sono numerosi ed impegnativi. Inoltre, novità rispetto al predecessore, si può gareggiare contemporaneamente in due: lo schermo apparirà diviso verticalmente per consentire ad entrambi i contendenti di vedere la parte di percorso circostante.
La grafica, per quanto semplice e priva di fronzoli, è discreta, le animazioni degli sprite delle auto sono realistiche e la giocabilità ottima. L'occasione ideale per un tuffo nel passato.

PREDATOR 2

Le software house non si lasciano sfuggire una sola occasione per realizzare giochi tratti da qualche film famoso: quasi in contemporanea con l'uscita sugli schermi cinematografici, la Mirrorsoft ha prontamente sfornato «Predator 2», tratto dal seguito del celebre film di fantascienza con Arnold Schwarzenegger.
Anche nel secondo episodio i terrestri sono alle prese con un gruppo di alieni dalla capigliatura simile a quella di Ruud Gullit e dalle perniciose abitudini venatorie: il loro

sport preferito consiste nel cacciare esseri umani. Il vostro compito, nei panni del tenente Mike Harrigan, è invece impedire che gli ignari abitanti di Los Angeles diventino trofei sul caminetto della casa di qualche Predator.
Il gioco, diviso in quattro livelli, è in pratica una versione riveduta e corretta di «Operation Wolf»: uno shoot'em up a scorrimento orizzontale visto in prospettiva nel quale dovete sparare praticamente a tutto ciò che si muove, evitando però di colpire gli innocenti passanti. I quattro livelli variano sostanzialmente solo nell'ambientazione e nella difficoltà: dovete eliminare gruppi di trafficanti di droga colombiani, in mezzo ai quali si annidano i Predator, a bordo di vagoni della metropolitana e per le strade della città, raccogliendo le munizioni e le nuove armi che appaiono di tanto in tanto.
La grafica e le animazioni sono ok, ma l'originalità è tristemente latente: se il film vi ha appassionato, potete togliervi la soddisfazione di tritare qualche alieno, restando comodamente seduti nella vostra poltrona.
Ma se già possedete una copia di «Operation Wolf» o del suo seguito, non vale la pena di spendere soldi per quello che sostanzialmente è un clone poco innovativo.

ELF

Sarebbe carino, una volta tanto, che il compito dell'eroe fosse quello di rapinare una banca o di strangolare la suocera o di investire un vigile urbano. Così, tanto per rompere con questa snervante routine di fidanzate rapite e di livelli a scrolling orizzontale pieni di troll e di bonus da raccogliere. E invece no: le software house continuano a propinare sempre le stesse trame e gli stessi meccanismi di gioco. Per fortuna la Ocean si fa perdonare, almeno per questa volta, sopperendo alla mancanza di orginalità con una buona dose di giocabilità ed un'iniezione di ottima grafica. In «Elf» l'eroe, un minuscolo elfo di nome Cornelius, deve attraversare sei livelli (composti da cento schermi ciascuno), inizialmente armato solo di una fionda.
Raccogliendo le armi e gli ingredienti disseminati lungo la strada, Cornelius può aumentare la propria potenza offensiva creando da sé alcuni incantesimi magici.
«Elf» richiede anche una certa dose di strategia e di ragionamento per superare certi ostacoli: informazioni utili possono essere acquisite fermandosi a discutere con alcuni dei personaggi che si incontrano lungo il cammino.

WANTED

SEI TU?

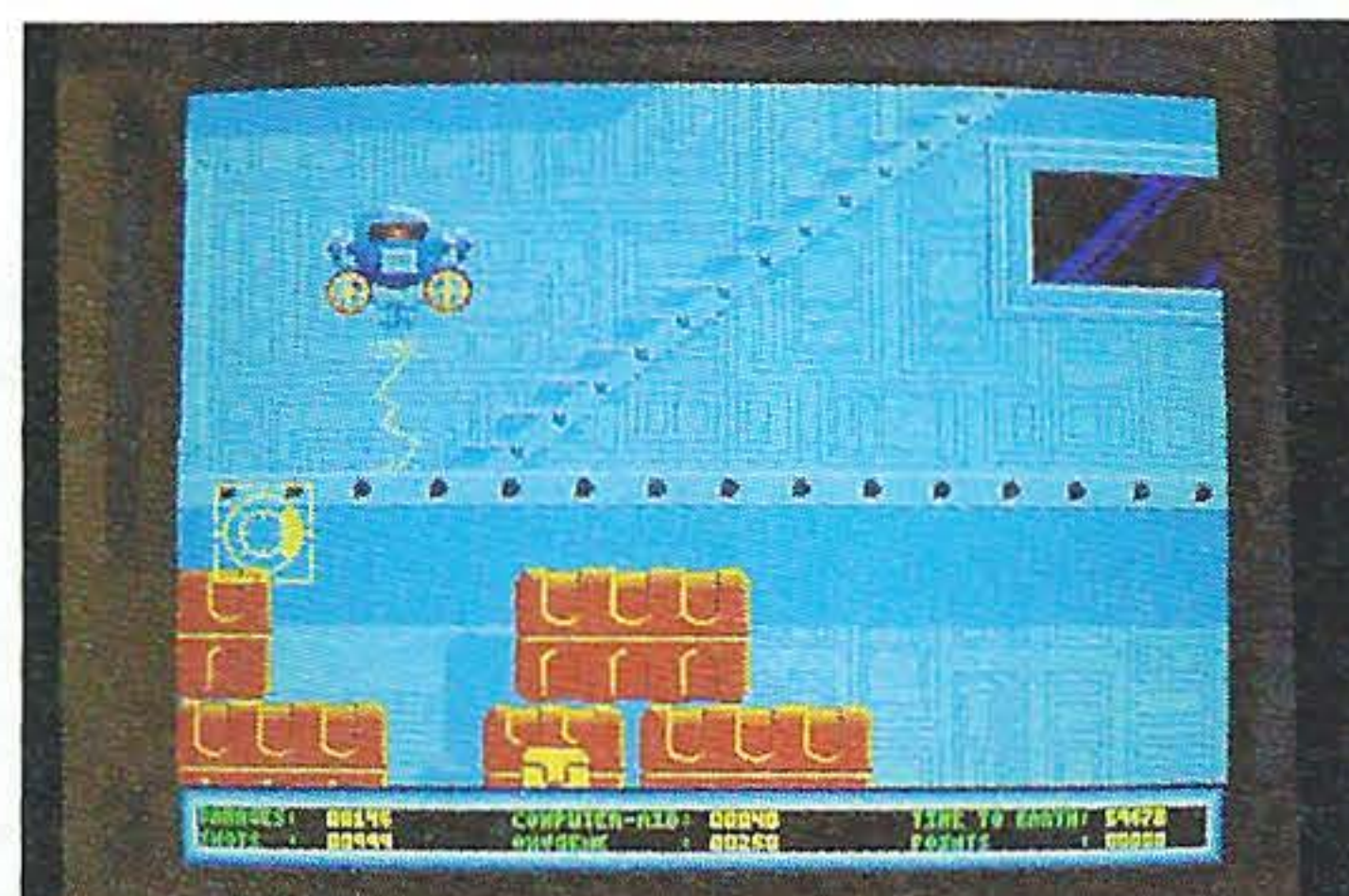
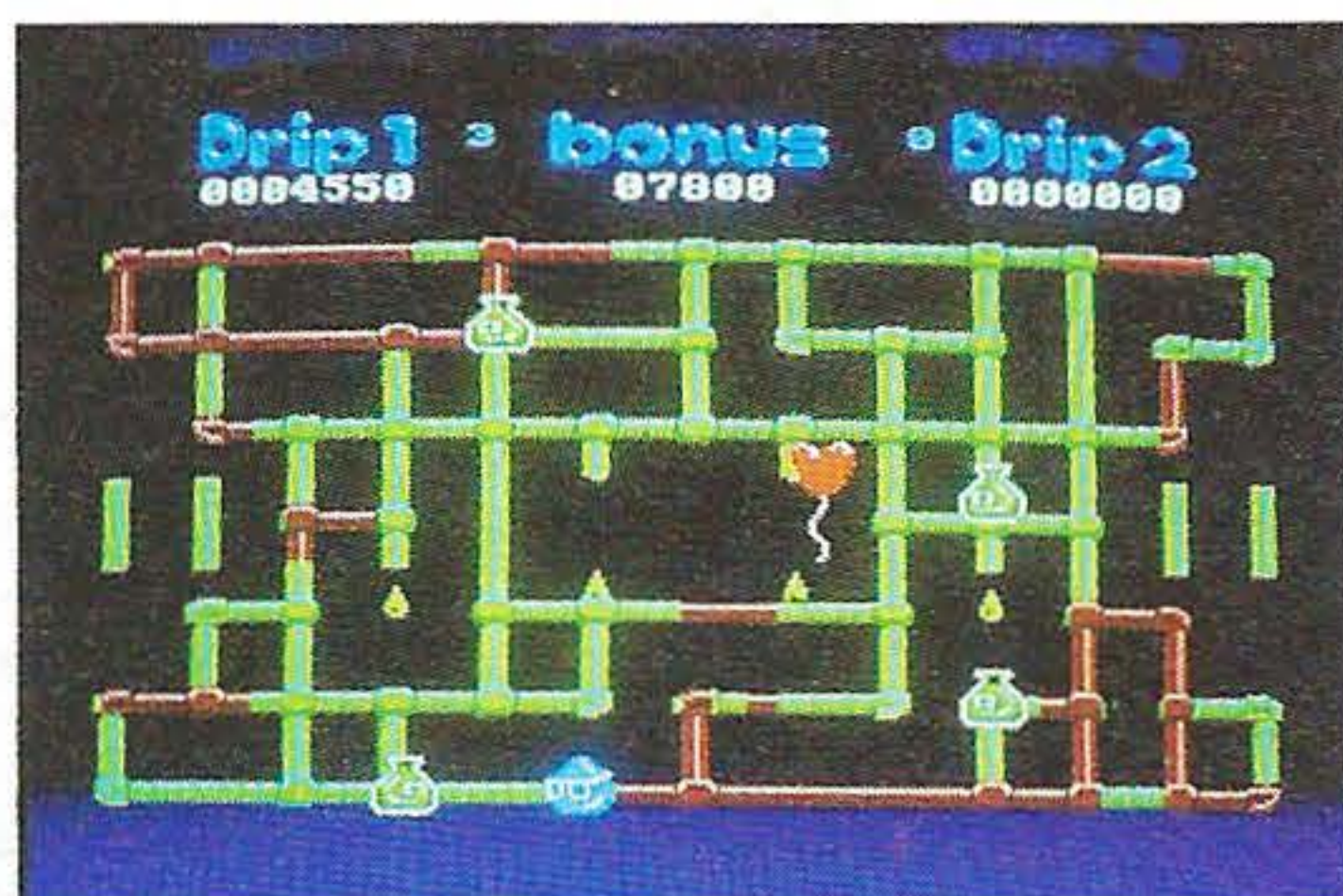
**SEI TU
IL PROSSIMO ABBONATO
AD
AMIGA BYTE**

**SOLO
LIRE 135.000**

Per 11 fascicoli
ed altrettanti dischetti
direttamente
a casa tua

IN PIÙ IN REGALO UN SUPERDISCO
CON DUE SPLENDIDI GIOCHI INEDITI

** Il superdisco viene inviato anche a chi si abbona a prezzo ridotto (L. 85mila) per ricevere esclusivamente i fascicoli senza dischetto allegato.*

Drip 1 bonus Drip 2

ABBONATI! Cosa aspetti?

**Per abbonarti invia vaglia postale ordinario ad Amiga Byte
c.so Vitt. Emanuele 15, 20122 Milano.**